Giambattista Amati, Claudio Carpineto, Giovanni Semeraro (Eds.)

Proceedings of the Third Italian Information Retrieval Workshop

IIR 2012

Department of Computer Science, University of Bari Aldo Moro, Italy
January 26-27, 2012
http://www.di.uniba.it/~swap/iir2012

This volume is published and copyrighted by:

Giambattista Amati

Claudio Carpineto

Giovanni Semeraro

ISSN 1613-0073

In memoriam of Barbara Asta, her 6 year old twins

Salvatore and Giuseppe and her husband Nunzio,

who could not stand the daily torment related to the

memory of Pizzolungo.

Table of Contents

Content Analysis

Applications

Preface

The purpose of the Italian Information Retrieval (IIR) workshop series is to provide an international meeting forum for stimulating and disseminating research in Information Retrieval and related disciplines, where researchers, especially early stage Italian researchers, can exchange ideas and present results in an informal way.

IIR 2012 took place in Bari, Italy, at the Department of Computer Science, University of Bari Aldo Moro, on January 26-27, 2012, following the first two successful editions in Padua (2010) and Milan (2011).

We received 37 submissions, including full and short original papers with new research results, as well as short papers describing ongoing projects or presenting already published results. Most contributors to IIR 2012 were PhD students and early stage researchers. Each submission was reviewed by at least two members of the Program Committee, and 24 papers were selected on the basis of originality, technical depth, style of presentation, and impact.

The 24 papers published in these proceedings cover six main topics: ranking, text classification, evaluation and geographic information retrieval, filtering, content analysis, and information retrieval applications. Twenty papers are written in English and four in Italian. We also include an abstract of the invited talk given by Roberto Navigli (Department of Computer Science, University of Rome "La Sapienza"), who presented a novel approach to Web search result clustering based on the automated discovery of word senses from raw text.

The Editors of the Conference Proceedings

Giambattista Amati
Fondazione Ugo Bordoni

Claudio Carpineto
Fondazione Ugo Bordoni

Giovanni Semeraro
University of Bari Aldo Moro

Organization

General Chair

- Giovanni Semeraro (University of Bari Aldo Moro)

Program Chairs

- Giambattista Amati (Fondazione Ugo Bordoni)
- Claudio Carpineto (Fondazione Ugo Bordoni)

Steering Committee

- Massimo Melucci (University of Padua)
- Stefano Mizzaro (University of Udine)
- Gabriella Pasi (University of Milano Bicocca)

Program Committee

- Giuseppe Amodeo (Fondazione Ugo Bordoni)
- Roberto Basili (University of Rome “Tor Vergata”)
- Marco Bianchi (Fondazione Ugo Bordoni)
- Gloria Bordogna (IDPA-CNR Dalmine, Bergamo)
- Fabio Crestani (Università della Svizzera Italiana, Lugano)
- Marco de Gemmis (University of Bari Aldo Moro)
- Pasquale De Meo (University of Messina)
- Giorgio Di Nunzio (University of Padua)
- Giorgio Gambosi (University of Rome “Tor Vergata”)
- Antonio Gulli (Search Technology Center, Bing Search, Microsoft)
- Monica Landoni (University of Strathclyde, Glasgow)
- Pasquale Lops (University of Bari Aldo Moro)
- Marco Maggini (University of Siena)
- Massimo Melucci (University of Padua)
- Alessandro Micarelli (University of Roma Tre)
- Stefano Mizzaro (University of Udine)
- Alessandro Moschitti (University of Trento)
- Roberto Navigli (University of Rome “La Sapienza”)
- Salvatore Orlando (University of Venice “Ca’ Foscari”)
- Gabriella Pasi (University of Milano Bicocca)
- Raffaele Perego (ISTI-CNR, Pisa)
- Francesco Ricci (Free University of Bozen-Bolzano)
- Fabrizio Sebastiani (ISTI-CNR, Pisa)
- Fabrizio Silvestri (ISTI-CNR, Pisa)

Organizing Committee

- Pierpaolo Basile (University of Bari Aldo Moro)
- Annalina Caputo (University of Bari Aldo Moro)
- Marco de Gemmis (University of Bari Aldo Moro)
- Leo Iaquinta (University of Bari Aldo Moro)
- Pasquale Lops (University of Bari Aldo Moro)
- Cataldo Musto (University of Bari Aldo Moro)
- Fedelucio Narducci (University of Bari Aldo Moro)

Acknowledgments

The workshop was supported by:

- Department of Computer Science of the University of Bari Aldo Moro
  `http://www.di.uniba.it`

- Distretto Produttivo dell'Informatica della Regione Puglia
  `http://www.distrettoinformatica.it`

- Informatici senza Frontiere
  `http://www.informaticisenzafrontiere.org`

and was sponsored by:

www.ethicasystem.com

www.exprivia.it

www.fub.it

www.linksmt.it

www.murexcs.it

www.nealogic.it

www.openworkbpm.com

www.questioncube.com

www.sudsistemi.it

Semantic is beautiful: clustering and diversifying search results with graph-based Word Sense Induction

Roberto Navigli

Department of Computer Science, Sapienza University of Rome
navigli@di.uniroma1.it

Abstract: Web search result clustering aims to facilitate information search on the Web. Rather than presenting the results of a query as a flat list, these are grouped on the basis of their similarity and subsequently shown to the user as a list of possibly labeled clusters. Each cluster is supposed to represent a different meaning of the input query, thus taking into account the language ambiguity issue. However, Web clustering methods typically rely on some notion of textual similarity of search results. As a result, text snippets with no word in common tend to be clustered separately, even if they share the same meaning.
In this talk, we present a novel approach to Web search result clustering based on the automatic discovery of word senses from raw text, a task referred to as Word Sense Induction (WSI). Key to our approach is to first acquire the senses (i.e., meanings) of a query and then cluster the search results based on their semantic similarity to the word senses induced. Our experiments, conducted on datasets of ambiguous queries, show that our approach outperforms both Web clustering and search engines in the clustering and diversification of search results.

Estensione dei metodi di ranking mediante analisi dell'interspaziatura fra occorrenze

Maria C. Daniele, Claudio Carpineto, and Andrea Bernardini

Fondazione Ugo Bordoni, Rome, Italy
mariac.daniele@gmail.com, carpinet@fub.it, aberna@fub.it

Abstract. L'analisi frequentistica delle occorrenze, tipica dei modelli di ranking di information retrieval, può essere integrata con l'analisi della spaziatura fra le occorrenze di una singola parola, mutuata dallo studio dei livelli di energia dei sistemi statistici di quanti disordinati. Queste due aree di ricerca sono fortemente interrelate, perché entrambe hanno l'obiettivo di assegnare dei pesi di rilevanza alle singole parole di un documento, e sembrano complementari, perché si basano su metodologie differenti. Tuttavia finora esse sono progredite in modo separato. L'obiettivo di questa ricerca è di favorire una loro riconciliazione. I contributi principali del lavoro sono tre: (a) estensione del metodo basato sull'interspaziatura mediante analisi di corpora, (b) verifica sperimentale che la pesatura quantistica è scorrelata da quella frequentistica, (c) studio della combinazione ottimale dei pesi quantistici e frequentistici ai fini del miglioramento delle prestazioni del ranking. Il risultato principale dei nostri esperimenti è che il metodo quantistico da solo non funziona bene, ma che il metodo combinato consente di migliorare in modo significativo le prestazioni del metodo classico frequentistico. Un ulteriore risultato riguarda le potenzialità di applicazione selettiva dei due metodi di pesatura: buone in funzione della lunghezza dei documenti recuperati, modeste rispetto alla difficoltà stimata delle interrogazioni.[1]

1 Introduzione

Ordinare i documenti di una collezione per pertinenza a fronte di una richiesta d'utente è il problema chiave dell'Information Retrieval. Nel corso degli ultimi decenni sono stati ideati numerosi modelli di ranking (vettoriale, probabilistico, basato sulla modellazione del linguaggio, o sullo scostamento dalla casualità), che tipicamente assegnano un punteggio o una probabilità a ciascun documento basandosi su una valutazione dell'importanza che i singoli termini dell'interrogazione rivestono nei documenti che li contiene. Le grandezze sulle quali si basano la maggior parte di questi modelli dipendono dalle frequenze con le quali i termini compaiono nei singoli documenti e nell'intera collezione. Coi

[1] Questo lavoro è basato sulla tesi di laurea magistrale in ingegneria informatica di Maria Daniele "Sperimentazione di tecniche d'Information Retrieval basate sulla Fisica dei Quanti", svolta presso la Fondazione Ugo Bordoni e discussa all'Università Roma Tre nel luglio 2011.

progressi degli ultimi anni però, i margini per ulteriori miglioramenti nelle tecniche tradizionali di ranking si sono ridotti: un avanzamento sostanziale ormai sarà difficile che avvenga senza un vero e proprio cambiamento di paradigma.

Parallelamente, nell'ultimo decennio si è sviluppato un ramo della ricerca riguardante l'estrazione delle parole rilevanti di un testo che prescinde dalla frequenza delle parole. Tale approccio, nato da studi sui livelli di energia dei sistemi statistici di quanti disordinati, si basa sull'analisi dell'interspaziatura fra le occorrenze di uno stesso termine. Un ruolo fondamentale è giocato dalle forze di attrazione e repulsione cui sono soggette le singole occorrenze di un termine. Più il termine è rilevante, maggiore è lattrazione fra sue occorrenze, quindi più tali parole si concentrano in aree determinate del documento, generando la formazione di clusters; viceversa, più un termine è comune e poco rilevante, più deboli sono queste forze, per cui il termine si distribuisce uniformemente lungo tutto il testo.

Ortuño et al. [6] sono stati i primi a mostrare che in un testo la distribuzione spaziale di una parola rilevante è molto diversa da quella corrispondente a una non rilevante, postulando un'analogia tra il linguaggio naturale e il linguaggio del DNA. In seguito, ci sono state altre proposte derivate da quella pioneristica di Ortuño et al., ad esempio [9], [5], e [1]. In [9] vengono accertate alcune limitazioni dell'indice di pesatura ideato da Ortuño et al. sulle quali ritorneremo in seguito. Una caratteristica importante di tutte queste tecniche quantistiche è che non serve una collezione esterna da analizzare: esse si basano esclusivamente sul contenuto dei singoli documenti.

Fra queste due aree di ricerca, quella frequentistica e quella quantistica, esiste una forte connessione, perché entrambe puntano ad assegnare un peso di rilevanza ai singoli termini di un documento. Tuttavia, esse sono state portate avanti in modo esclusivo nelle due comunità, di information retrieval e di fisica dei quanti, senza cercare di analizzare i rispettivi vantaggi e svantaggi o di combinarle per trovare un approccio più potente di quelli singoli. Da questa osservazione è scaturita la nostra ricerca. L'obiettivo è il tentativo di cominciare a riconciliare questi due approcci.

La prima area di intervento è stata l'estensione della pesatura quantistica con statistiche estratte da un corpus (considerando in particolare le variazioni della frequenza di ciascun termine rispetto all'insieme dei documenti), ai fini di premiare la capacità di discriminazione di un termine. Il secondo tema che è stato studiato è la complementarietà dei ranking prodotti dalle metriche quantistiche e da quelle frequentistiche (in particolare quelle basate su tf-idf). Infine, dati gli esiti deludenti dell'applicazione diretta delle metriche quantistiche, con o senza estensione, al ranking dei documenti, abbiamo fatto una serie di esperimenti per valutare l'efficacia di una combinazione dei due metodi. I risultati sono stati incoraggianti, con prestazioni migliori di quelle ottenibili con i metodi convenzionali di information retrieval (in particolare BM25).

Il seguito di questo articolo è strutturato nel seguente modo. Dopo avere ricapitolato l'approccio quantistico alla pesatura dei termini, così come presentato in letteratura, introduciamo la sua estensione basata sulle variazioni di frequenza

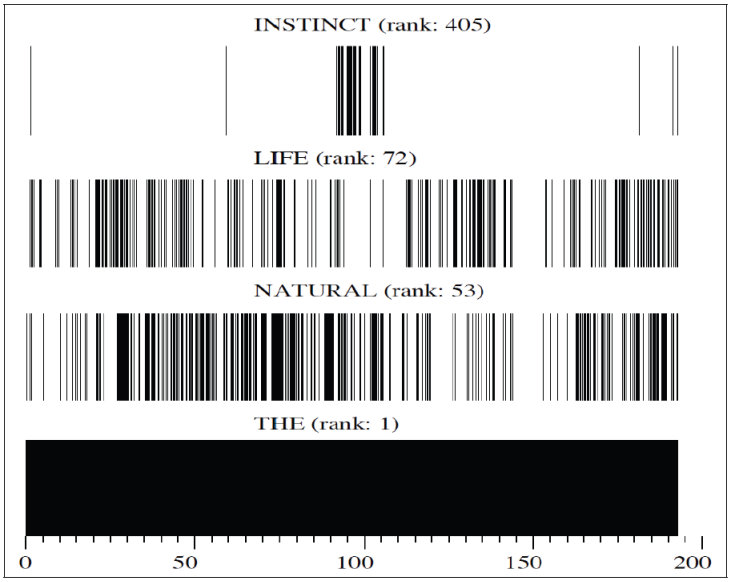


Fig. 1. Analisi spettrale e rank dei termini istinct, life, natural e the, estratte dal testo di Charles Darwin The Origin of Species" [6].

nel corpus. Successivamente, viene discussa la combinazione di pesatura quantistica estesa e pesatura tradizionale ai fini del ranking, presentando una serie di esperimenti su due collezioni campione. Infine, viene discussa la possibilità di un uso selettivo delle due metriche di pesatura guidato da lunghezza dei documenti e difficoltà delle interrogazioni.

2 Pesatura delle parole basata su interspaziatura fra occorrenze: σ_p

Il fenomeno della diversa distribuzione spaziale di parole rilevanti e non rilevanti è illustrato in Figura 1. I grafici sono relativi al testo The Origin of Species di Charles Darwin. Le occorrenze di parole rilevanti, come istinct, natural, e life, hanno distribuzione non omogenea e tendono a unirsi (fenomeno d'attrazione) formando dei clusters. Ciò accade indipendentemente dal numero di occorrenze, perché queste parole hanno nel ranking delle frequenze posizioni differenti. Simmetricamente, le occorrenze di parole non rilevanti, quali the (che è il termine con maggiore frequenza nel testo), sono equidistribuite.

Da un punto di vista fisico, nel caso di una parola chiave ogni livello d'energia attrae se stesso. La controparte linguistica di questo comportamento è che un termine rilevante è di solito il soggetto principale in un contesto locale di un documento, perciò occorre con maggiore frequenza in qualche area del testo e minore

in altre, generando il fenomeno di clustering. Invece, nel caso di parole non rilevanti tali livelli d'energia risultano scorrelati, corrispondentemente al fatto che tali parole si distribuiscono attraverso l'intero documento senza caratterizzarne in modo specifico nessuna parte.

Per quantificare questo fenomeno si utilizza il seguente approccio. Ogni occorrenza di un termine è considerata come un livello di energia che si trova all'interno di uno spettro energetico formato da tutte le occorrenze della data parola nel testo che si sta analizzando. Ogni valore del livello di energia è dato semplicemente dalla posizione che il termine ha nel documento. In pratica, per una data parola w, si estraggono le posizioni corrispondenti, creando il vettore x(w) = x_1 ,..., x_n (ogni x_i corrisponde ad un livello di energia). Ad esempio, nella frase "a great scientist must be a good teacher and a good researcher", per la parola "a" si estrae il vettore di posizioni x(a) =1,6,10. Si considera, poi, il vettore delle distanze d_i, dist(w) = d_1,...,d_n, con $d_i = x_{i+1}$ - x_i, tra le occorrenze consecutive della parola w e si calcola la corrispondente media delle distanze μ:

$$\mu \;=\; \frac{1}{n+1} \;\cdot\; \sum_{i=0}^{n} (x_{i+1} - x_i) \;=\; \frac{x_{n+1} - x_0}{n+1} \tag{1}$$

Denotando con p(x) la frequenza relativa di occorrenza di una data distanza x, la sua funzione di distribuzione integrata P_1(x) è:

$$P_1(x) \;=\; \sum_{x^{'}<x} p(x^{'}) \tag{2}$$

Se la parola è distribuita in modo casuale (random) lungo il testo, la distribuzione P_1, nel limite continuo, sarà una distribuzione Poissoniana:

$$P_1(\mu) \;=\; 1 - exp(-\mu) \tag{3}$$

Se invece il termine respinge se stesso (quindi è distribuito uniformemente lungo tutto il testo) allora la sua distribuzione P_1 sarà più piccola di quella di Poisson per $\mu < 1$. Viceversa, se il termine attrae se stesso, P_1 sarà più grande della distribuzione di Poisson per brevi distanze (per un trattamento probabilistico piu approfondito si rimanda a [6]). Questo perché, come già osservato, le parole rilevanti di un testo compaiono generalmente in un ambito specifico, con oscillazioni apprezzabili fra i diversi ambiti.

Il calcolo della funzione di distribuzione P_1 per tutte le parole di un testo è molto oneroso dal punto di vista computazionale. Per questo motivo, al posto di P_1, viene utilizzata la deviazione standard s:

$$s \;=\; \frac{1}{n-1} \;\cdot\; \sum_{i=0}^{n} ((x_{i+1} - x_i) - \mu)^2 \tag{4}$$

Per eliminare la dipendenza dalla frequenza per differenti parole, la deviazione standard viene normalizzata rispetto al corrispondente valore medio delle distanze moltiplicato per $\sqrt{1-p}$:

$$\sigma_p \;=\; \frac{s}{\mu} \;\cdot\; \frac{1}{\sqrt{1-p}} \tag{5}$$

dove n è il numero di occorrenze della parola w all'interno del documento e N è il numero totale di parole nel testo. Questa funzione è molto semplice da calcolare e si dimostra robusta contro le oscillazioni. Le parole con il valore di σ_p più elevato saranno quelle più importanti.

3 Estensione della pesatura quantistica mediante analisi di corpora: σ^*

Numerose analisi e modifiche di σ_p sono state proposte. Uno dei lavori piu importanti è [9], dove sono evidenziati vari problemi. Il primo è che può accadere che parole comuni (rilevanti) abbiano alto (basso) valore di σ_p. Ad esempio, la parola you, che è indubbiamente un termine con scarso valore informativo, nella Bibbia ha valore 2,71 ed è classificata in posizione 550, che è molto elevata considerando che ci sono 12.910 parole distinte all'interno del libro; inoltre, la parola Sirach rispetto a you è più rilevante, ma ha solo un valore pari a 0,24 con corrispondente ranking di 9543. In secondo luogo, il metodo è alquanto instabile perché il valore di σ_p puo' essere influenzato fortemente dal cambio di una delle posizioni, specialmente in testi molto grandi. Ancora, ad alti valori non sempre corrisponde una distribuzione concentrata localmente. Ad esempio, la distribuzione 3,5,7,20 è clusterizzata nella regione [3,7], mentre per 3,5,18,20 si trovano due piccoli cluster in [3,5] e [18,20]; la metrica non fa distinzione tra questi due insiemi, a cui corrisponde lo stesso valore di σ_p. Un altro problema evidenziato, particolarmente importante per la nostra applicazione, è che la dimensione di un testo ha un forte impatto sulle prestazioni generali del sistema. Più il testo è breve, più l'indice classifica male le parole, collocando fra le prime posizioni quelle parole con frequenze molto basse, che all'interno del documento compaiono solamente pochissime volte e in posizioni molto ravvicinate tra loro (che in testi corti può accedere anche ad articoli o preposizioni).

I tentativi presenti in letteratura hanno cercato di presentare dei correttivi alla funzione σ_p senza però abbandonare l'assunzione di base, e cioè che l'ordinamento dei termini viene costruito soltanto analizzando il particolare testo che si sta considerando. Mentre questa assunzione può essere utile in determinate situazioni, sembra ragionevole cercare di estendere l'approccio quantistico utilizzando informazioni aggiuntive sulla importanza dei singoli termini basate sull'analisi di corpora, la disponibilità di corpora essendo oggigiorno vasta.

In particolare, noi proponiamo di correggere la metrica originaria con un fattore che abbia un duplice obiettivo: penalizzare le parole rare, perché in collezioni reali queste spesso costituiscono "rumore", e premiare le parole che riescono a discriminare meglio il testo in osservazione da altri testi, capacità questa che manca completamente nella pesatura quantistica. Il nostro approccio prende lo spunto da una metrica ben nota in information retrieval, la deviazione standard delle frequenze dei termini [7] Essa è definita nel seguente modo. Si consideri il

vettore delle frequenze f_i, freq(w) = $f_1, ..., f_{ND}$ relativo a una ad una parola w negli ND documenti della collezione., La media delle frequenze μ_f è:

$$\mu_f(w) \ = \ \frac{1}{ND} \ \cdot \ \sum_{i=1}^{n} \ f_i(w) \tag{6}$$

Si noti che ND è il numero totale di documenti della collezione: vengono considerate quindi anche le frequenze nulle, cioè i documenti in cui la parola non compare. La deviazione standard delle frequenze s_f sara' data da:

$$s_f(w) \ = \ \frac{1}{ND} \ \cdot \ \sum_{i=1}^{n} \ (f_i(w) - \mu_f)^2 \tag{7}$$

Chiaramente, s_f assumerà valori piccoli nel caso in cui la distribuzione di frequenza è uniforme (con f_i circa uguale a μ_f) o il termine appare in pochissimi documenti (essendo la maggior parte degli f_i uguali a zero e μ_f circa uguale a zero). Viceversa, s_f sarà grande quando la distribuzione di frequenza presenta forti variazioni a fronte di una frequenza media apprezzabile. Queste caratteristiche sembrano in grado di compensare i limiti di σ_p.

La deviazione standard può essere poi normalizzata rispetto al corrispondente valore medio delle frequenze μ_f, come visto in precedenza nel caso della pesatura quantistica:

$$\sigma_f \ = \ \frac{s_f}{\mu_f} \tag{8}$$

Nel complesso, questo approccio ha l'ulteriore vantaggio che il suo razionale è analogo a quello impiegato per sviluppare la funzione di pesatura originale σ_p. In questo caso i livelli di energia di una parola non corrispondono più alla posizione delle sue occorrenze in un testo, bensì alle frequenze in ciascun documento della collezione. Pertanto, l'analogia in questo caso è fra lo spettro di energia dei sistemi di quanti disordinati e l'insieme delle frequenze che una certa parola assume nella collezione.

La funzione di pesatura quantistica estesa σ^*, relativa ad una singola parola, è data dal prodotto di σ_p e σ_f:

$$\sigma^*(w) \ = \ \sigma_p(w) \ \cdot \ \sigma_f(w) \tag{9}$$

Per farsi un'idea più precisa delle caratteristiche dei termini estratti da testi lunghi mediante metriche frequentistiche e quantistiche, nonché del loro grado di complementarietà, abbiamo svolto il seguente esperimento. Come metrica di pesatura frequentistica abbiamo scelto tf-idf, che è semplice ed ha una valenza paradigmatica in information retrieval, nelle due versioni con e senza stop words (denotate rispettivamente tf-idf e tf-idf*), e come metriche quantistiche σ_p e σ^*. Abbiamo utilizzato come testo The Bible[2] e come corpus di riferimento per

[2] http://www.gutenberg.org/ebooks/10

calcolare i valori tf-idf e σ_f la collezione TREC WT10g, pre-elaborarata secondo quanto descritto nella Sezione 5.

I risultati sono mostrati in Tabella 1. La metrica tf-idf ha riportato nelle prime posizioni molte stop words arcaiche, poiché queste parole, oltre ad avere un valore elevato di tf nel testo originario, hanno conseguito anche un alto valore di idf nella collezione di riferimento (costituita da testi moderni). La metrica tf-idf* (cioè con rimozione di stop words) ha funzionato molto meglio, anche se ha restituito diversi termini generici nelle prime dieci posizioni, quali ad esempio "son", "king", "man", "land", "men". Le parole estratte da σ_p sembrano invece piú precise nel descrivere il contenuto della Bibbia, e consentono di identificare molti concetti e nomi propri importanti. Passando a σ^*, si nota che le parole diventano ancora piú specifiche (anche se non si tratta di termini rari in un testo come la Bibbia) e corrispondono a brani piú circoscritti all'interno del libro. Alcuni di questi termini hanno conseguito un alto valore di σ^* non solo in virtú della loro elevata concentrazione nella Bibbia ma anche per l'infrequenza con la quale appaiono nel corpus, secondo quanto giá evidenziato nella discussione di tf-idf. Nel complesso le parole estratte da σ^* sono meno caratterizzanti al livello del testo globale ma hanno sicuramente una maggiore capacità di discriminazione (ad esempio rispetto ad altri testi di carattere religioso).

rank	tf-idf	tf-idf*	σ_p	σ^*
1	unto (1,14)	lord (6,64)	jesus (24,35)	jesus (7,89)
2	shall (0,82)	god (3,12)	christ (18,31)	saul (4,97)
3	lord (0,81)	absalom (2,287)	paul (11,74)	absalom (4,97)
4	thou (0,71)	son (1,74)	peter (9,91)	jephthah (2,08)
5	thy (0,60)	king (1,55)	disciples (9,64)	jubile (2,08)
6	thee (0,50)	behold (1,46)	faith (9,39)	ascendeth (2,07)
7	him (0,42)	man (0,40)	john (9,14)	abimelech (1,96)
8	god (0,38)	judah (1,10)	david (8,75)	elias (1,95)
9	his (0,38)	land (1,05)	saul (8,70)	joab (1,86)
10	hath (0,31)	men (1,02)	gospel (8,01)	haman (1,82)

Table 1. Ordinamento e punteggi dei primi dieci termini della Bibbia secondo le metriche tf-idf (con e senza stop words), σ_p, e σ^*.

Se poi confrontiamo la somiglianza dei ranking prodotti dalle diverse metriche, ci accorgiamo che metriche frequentistiche e quantistiche restituiscono termini molto differenti. Considerando i primi 100 termini, ci sono 15 termini in comune fra σ_p e i due tf-idf, che scendono a due con σ^*, precisamente "jesus" e "saul". Inoltre, i pochi termini in comune hanno posizioni molto differenti. Ad esempio, la parola "jesus", che usando σ_p e σ^* compare nella prima posizione, viene invece classificata rispettivamente in quarantesima e quindicesima posizione da tf-idf e tf-idf*. Questi risultati indicano chiaramente che i ranking prodotti dai due tipi di ordinamento sono completamente scorrelati, in particolar modo quando si considera σ^* invece di σ_p, anche se bisogna sottolineare che i

nostri esperimenti sono stati effettuati su un testo lungo che non contiene errori. I testi che vengono tipicamente considerati nelle applicazioni di information retrieval sono invece brevi e rumorosi. Nelle prossime sezioni verranno presentati una serie di esperimenti con la seconda tipologia di dati.

4 Applicazione di σ^* al ranking

La metrica σ^* puo essere adoperata per fare il ranking di una collezione di documenti rispetto ad una interrogazione q, semplicemente sommando i valori relativi a tutti i termini di q presenti nel documento. Il punteggio $\sigma^*(d, q)$ conferito al generico documento d sarà dato da:

$$\sigma^*(d, q) \ = \ \sum_{w \in q} \sigma^*(w) \tag{10}$$

Vista la complementarietà delle metriche di pesatura quantistica e frequentistica, un approccio naturale è quello di cercare di integrare le due tecniche. Uno dei modi più intuitivi è fare una combinazione lineare dei punteggi assegnati dalle due tecniche a ciascun documento, preceduta da una normalizzazione degli stessi. Lo schema di normalizzazione adoperato è stato il seguente:

$$weight_{NORM} \ = \ \frac{weight - weight_{Min}}{weight_{Max} - weight_{Min}} \tag{11}$$

Il punteggio finale è dato da:

$$score \ = \ \alpha \cdot score_{BM25} \ + \ (1 - \alpha) \cdot score_{\sigma^*} \tag{12}$$

5 Esperimenti

Come collezioni di prova abbiamo utilizzato la WT10g e la Robust, due collezioni sviluppate in ambito TREC. La prima contiene oltre un milione e mezzo di pagine web, la seconda circa 500 mila documenti estratti da varie sorgenti informative. Per WT10g sono state utilizzate le 50 topics 501-550, mentre per la collezione Robust sono state usate 250 queries, le topics 301-450 che sono quelle del track "ad hoc" delle TREC 6-8, e le topics 601-700 del track " robust" delle TREC 2003-2004. Su queste collezioni è stata applicata una riduzione dello spazio dei termini, sia per rendere pi ù efficiente l'esecuzione degli esperimenti sia per cercare di migliorare l'efficacia attraverso una riduzione del rumore insito nei testi (abbreviazioni, refusi, ecc.). In particolare sono state rimosse le parole contenute in meno di dieci documenti, e quelle che contenevano pi ù di tre caratteri consecutivi uguali o che erano lunghe pi ù di venti caratteri. Tale procedimento ha portato l'insieme di documenti WT10g ad avere 435.744 invece di 5.167.898 di termini distinti (considerando anche i numeri interi), mentre per la Robust siamo passati da 1.178.484 a 485.326. Per quest'ultima collezione però abbiamo notato che per alcune topics c'era soltanto un documento che conteneva i

termini corrispondenti; eliminando la restrizione sulla frequenza dei documenti siamo passati a 835.760 termini.

Come sistema di indicizzazione e ricerca è stato utilizzato Lucene,[3] con l'estensione a BM25 fornita da Perez-Iglesias[4]. Lucene è stato adoperato sia per calcolare il ranking secondo BM25, sia per fornire i documenti di input (tutti quelli che contenevano almeno una parola dell'interrogazione) alle routine sviluppate per calcolare il ranking secondo σ^* e il successivo ranking integrato σ^* + BM25. Il valore di α usato negli esperimenti (= 0,8) è stato determinato utilizzando le topics 451-500, viste come al training set di WT10g.

In Tabella 2 sono riportate le prestazioni dei tre metodi di ranking, cioè BM25, σ^* e la loro combinazione BM25+σ^*, su ciascuna delle due collezioni.[5] BM25 va molto meglio di σ^*, probabilmente a causa del fatto che i documenti rilevanti sono di lunghezza ridotta, ma il metodo combinato ha ottenuto le prestazioni migliori in tutti e due i casi, con un miglioramento piuttosto netto anche rispetto a BM25. La differenza fra le prestazioni del metodo integrato e di BM25 sono statisticamente significative utilizzando il T-Test.

Collezione	Topics	BM25	σ^*	BM25 + σ^*
WT10g	501-550	0.143	0.057	0.153
Robust	301-450, 601-700	0.195	0.089	0.203

Table 2. MAP (mean average precision) medio dei metodi di ordinamento singoli e combinati sulle collezioni WT10g e Robust.

Per esaminare meglio le prestazioni relative dei tre metodi, abbiamo calcolato il valore di MAP sulle singole interrogazioni. In Figura 2 abbiamo graficato i risultati per le interrogazioni di WT10g. In questo caso il metodo combinato migliora in 28 casi e peggiora nei rimanenti 22, rispetto a BM25. I risultati per Robust sono leggermente differenti, perché a fronte di un miglioramento medio percentuale piu contenuto, la robustezza rispetto alle singole interrogazioni aumenta: 197 i miglioramenti, 53 i peggioramenti.

Per valutare la robustezza del metodo rispetto al parametro α abbiamo ricalcolato le prestazioni facendo variare il valore di α nell'intervallo fra uno e zero, i due estremi coincidendo rispettivamente con BM25 e σ^*. I risultati, mostrati in Tabella 3, suggeriscono chiaramente che il metodo è sufficientemente robusto, perché c'è un intervallo di valori per i quali le prestazioni si mantengono elevate, e questo comportamento è riscontrabile su entrambe le collezioni.

[3] http://lucene.apache.org/

[4] http://nlp.uned.es/ jperezi/Lucene-BM25/

[5] Abbiamo fatto una serie di esperimenti per valutare le potenzialità per il ranking anche della metrica σ_p, sia da sola, sia in combinazione con BM25, sia infine come riordinamento del ranking prodotto da BM25. I risultati però sono stati insoddisfacenti.

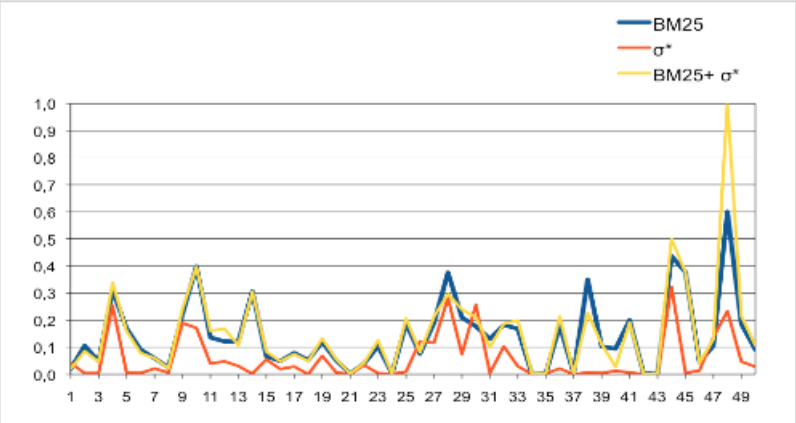


Fig. 2. Analisi delle prestazioni sulle singole topics di WT10g.

α	1	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2	0.1	0
WT10g	0.143	0.146	0.153	0.153	0.150	0.137	0.122	0.096	0.081	0.067	0.054
Robust	0.195	0.203	0.203	0.198	0.167	0.154	0.142	0.120	0.107	0.096	0.089

Table 3. MAP medio del metodo di ranking combinato BM25+σ^* sulle due collezioni, al variare del parametro α.

6 Applicazione selettiva delle metriche frequentistiche e quantistiche

Finora abbiamo considerato l'ipotesi di combinare la pesatura frequentistica e quantistica in modo sistematico, per ciascuna interrogazione e su tutta la collezione. Poiche però pesatura quantistica e frequentistica hanno caratteristiche e requisiti differenti, ci siamo chiesti se è possibile prevedere una utilizzazione selettiva dei due paradigmi di ranking in funzione di determinate caratteristiche dei documenti e dell'interrogazione. La prima variabile che abbiamo considerato è stata la lunghezza dei documenti, perché il metodo quantistico dovrebbe andare meglio sui testi lunghi. Vogliamo valutare se effettivamente la metrica quantistica è più efficace nel recuperare i documenti lunghi e quella frequentistica i documenti brevi.

A questo scopo abbiamo riportato due grafici relativi a WT10g, uno per BM25 e uno per σ^*, in cui sull'asse x ci sono i valori della lunghezza del documento in numero di parole, mentre sull'asse y è riportata la percentuale di documenti rilevanti (nei due casi in cui vengano ritrovati o non ritrovati) che hanno meno del corrispondente numero di parole dell'asse x. Ad esempio, il grafico di sinistra mostra che per i documenti rilevanti di lunghezza < 2000, i ritrovati da BM25 sono l80% del totale dei rilevanti ritrovati e solo il 60% dei rilevanti non ritrovati. Risulta quindi confermato che gli andamenti sono opposti

a seconda della metrica che si considera. Questi risultati sono incoraggianti dal punto di vista di un'applicazione selettiva guidata dalla lunghezza dei documenti. Lo sviluppo e la sperimentazione di un metodo di pesatura basato su queste osservazioni è stato lasciato come lavoro futuro.

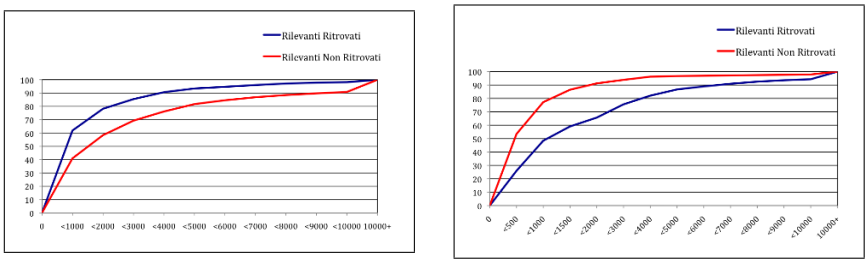


Fig. 3. Percentuali cumulative dei documenti rilevanti ritrovati e rilevanti non ritrovati da BM25 (sinistra) e σ^* (destra), in funzione della lunghezza dei documenti.

La seconda variabile per l'applicazione selettiva che abbiamo considerato è stata la difficoltà stimata delle interrogazioni. La speranza era che le metriche fossero efficaci in modo inverso rispetto a quest'ultima, in particolare che la pesatura quantistica conseguisse buone prestazioni sulle topics ritenute piu difficili. Abbiamo utilizzato due noti predittori pre-retrieval: Simplified Clarity Score [4] e σ_1 [8]. In Figura 4 abbiamo riportato due grafici, uno per WT10g con predittore σ_1 e uno per Robust con predittore Simplified Clarity Score, in cui ciascuna topic viene rappresentata con il valore restituito dal predittore (asse x) e con il suo valore di MAP (asse y), quest'ultimo calcolato utilizzando sia BM25 sia a σ^*. Nelle figure sono graficate anche le rispettive regressioni lineari. Risulta chiaro che le due metriche hanno un comportamento simile. In questo caso quindi, non sembrano esserci i presupposti per un'applicazione selettiva delle due tecniche.

7 Conclusioni

In questo lavoro abbiamo cercato di riconciliare la pesatura quantistica delle parole, basata sull'interspaziatura delle occorrenze e sviluppata prevalentemente nell'ambito della fisica, e la pesatura frequentistica adottata in information retrieval. Abbiamo visto che le due tecniche sono essenzialmente complementari e che la loro combinazione può migliorare sia la pesatura quantistica, incorporando statistiche legate all'analisi di corpus, sia quella frequentistica, per trovare termini rilevanti che sfuggono ai normali criteri basati su tf-idf. In una serie di esperimenti preliminari abbiamo dimostrato che è possibile migliorare il ranking attraverso una semplice combinazione delle due metriche, anche se le potenzialità di questo approccio sono ancora in gran parte da investigare. Oltre al ranking, questa tecnica può essere utilizzata per migliorare altri classici compiti di information retrieval nei quali l'individuazione delle parole chiave presenti in uno

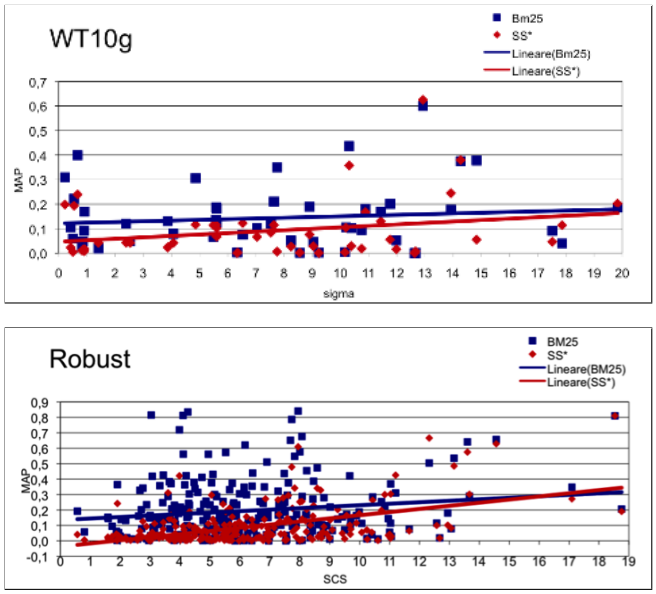


Fig. 4. MAP delle singole topics in funzione della loro difficolta' stimata

o più documenti è cruciale ed è stata finora affrontata con tecniche frequentistiche, in particolare la diversificazione e il clustering dei risultati [2] e l'espansione automatica delle interrogazioni [3].

References

1. P. Carpena, P. Bernaola-Galvă, M. Hackenberg, A. V. Coronado, and J. L. Oliver. Level statistics of words: Finding keywords in literary texts and symbolic sequences. *Physical Review E 79:035102*, 2009.
2. C. Carpineto, M. D'Amico, and G. Romano. Evaluating Subtopic Retrieval Methods: Clustering Versus Diversification of Search Results. Information Processing and Management, in press, 2012.
3. C. Carpineto and G. Romano. A Survey of Automatic Query Expansion in Information Retrieval . ACM Computing Surveys, in press, 2012.
4. B. He and I. Ounis. Query performance prediction. *Inf. Sys.*, 31(7):585–594, 2006.
5. J. P. Herrera and P. A. Pury. Statistical keyword detection in literary corpora. *European Physical Journal B*, 63:135–146, 2008.
6. M. Ortuño, P. Carpena, P. Bernaola-Galvă, E. Muñoz, and M. Somoza. Keyword detection in natural languages and dna. *Europhysics Letters*, 57(5):759–764, 2002.
7. G. Salton. *A Theory of indexing*. Society for Industrial and Applied Mathematics, 1975.
8. Y. Zhao, F. Scholer, and Y. Tsegay. Effective pre-retrieval query performance prediction using similarity and variability evidence. In *ECIR '08*, pages 52–64, 2008.
9. H. Zhou and G. W. Slater. A metric to search for relevant words. *Phyisica A 329*, pages 309–327, 2003.

Orthogonal negation for document re-ranking*

Pierpaolo Basile, Annalina Caputo, and Giovanni Semeraro

Dept. of Computer Science - University of Bari "Aldo Moro"
Via Orabona, 4 - I-70125, Bari (ITALY)
basilepp@di.uniba.it, acaputo@di.uniba.it, semeraro@di.uniba.it

Abstract. In this work, we propose a method for document re-ranking, which exploits negative feedback represented by non-relevant documents. The concept of non-relevance is modelled through the quantum negation operator. The evaluation carried out on a standard collection shows the effectiveness of the proposed method in both the classical Vector Space Model and a Semantic Document Space.

1 Introduction

This work investigates the role of non-relevant documents in document re-ranking. Classic relevance feedback methods are able to handle negative feedback by subtracting "information" from the original query. However, these approaches suffer from the side effect caused by information loss. To deal with this effect, we propose a negative feedback based on quantum negation that is able to remove only the unwanted aspects pertaining to non-relevant documents. The key idea behind our approach is to build a document vector d^* corresponding to an *ideal document* which best fits the user's need, and then re-rank the initial set of ranked documents D_{init} by computing the similarity between d^* and each document in D_{init}. The ideal document vector d^* should fit the *concepts* in the set of relevant documents D^+, while skipping *concepts* in the set D^- of non-relevant ones. Formally, a new relevance score is computed for each document $d_i \in D_{init}$ according to the following equation:

$$S(d_i) = \alpha * S_{D_{init}}(d_i) + (1 - \alpha) * sim(d_i, d^*) \quad (1)$$

where $S_{D_{init}}(d_i)$ is the score of d_i in the initial rank D_{init}, while $sim(d_i, d^*)$ is the similarity degree between the document vector d_i and the ideal document vector d^* computed by cosine similarity. The outcome of the process is a list of documents ranked according to the new scores computed using Equation 1. In our approach, documents are represented as vectors in a geometric space in which similar documents are represented close to each other. This space can be the classical *Vector Space Model* (VSM) or a *Semantic Document Space* (SDS)

* This paper summarizes the main results already published in Basile, P., Caputo, A., Semeraro, G.: Negation for document re-ranking in ad-hoc retrieval. In: Amati, G., Crestani, F. (eds.) Advances in Information Retrieval Theory, Lecture Notes in Computer Science, vol. 6931, pp. 285–296. Springer Berlin / Heidelberg (2011)

induced by a distributional approach. Moreover, we compare our strategy with a classical strategy based on "information subtraction".

2 Re-ranking using quantum negation

To build the ideal document d^* we use a geometrical space where d^* is computed as a vector close to relevant documents and unrelated to non-relevant ones. In our space the concept of relevance is expressed in terms of similarity, while the concept of irrelevance is defined by orthogonality (similarity equals to zero). Formally, we want to compute the vector which represents the following logical operation:

$$d^* = d_1^+ \vee d_2^+ \vee \ldots \vee d_n^+ \wedge NOT(d_1^-) \wedge NOT(d_2^-) \wedge \ldots \wedge NOT(d_m^-) \quad (2)$$

where $D^+ = \{d_i^+, i = 1 \ldots n\}$ and $D^- = \{d_j^-, j = 1 \ldots m\}$ are the subsets of relevant and non-relevant documents respectively.

As shown in [5], given two vectors a and b in a vector space V endowed with a scalar product, $a \quad NOT \quad b$ corresponds to the projection of a onto the orthogonal space $\langle b \rangle^\perp \equiv \{v \in V : \forall b \in \langle b \rangle, v \cdot b = 0\}$, where $\langle b \rangle$ is the subspace $\{\lambda b : \lambda \in \mathbb{R}\}$. Equation 2 consists in computing a vector which represents the disjunction of the documents in D^+, and then projecting this vector onto all m orthogonal spaces defined by the documents in D^-. This operation is quite complex to compute, but applying De Morgan rules to the conjunction of negations, it can be transformed in a single negation of disjunctions:

$$d^* = d_1^+ \vee d_2^+ \vee \ldots \vee d_n^+ \wedge NOT(d_1^- \vee d_2^- \vee \ldots \vee d_m^-) \quad (3)$$

Thus, it is possible to build the ideal document vector d^* in two steps:

1. compute the disjunction of relevant documents as the vector sum of relevant documents. Indeed, disjunction in set theory is modelled as set union, which corresponds to the vector sum in linear algebra;
2. compute the projection of the vector sum of relevant documents onto the orthogonal space defined by the vector sum of non-relevant documents, for example using the Gram-Schmidt method. This means that the result vector captures those aspects that are common to relevant documents and are distant from non-relevant ones.

Disjunction and negation using quantum logic are thoroughly described in [5]. An overview of Quantum Mechanics for Information Retrieval can be found in [2]. Finally, the re-ranking algorithm is performed by computing the Equation 1.

3 Evaluation and Remarks

The aim of our evaluation is twofold. We want to prove that our re-ranking strategy based on quantum negation improves retrieval performance and outperforms the "information subtraction" method. To perform re-ranking using

a classical "information subtraction" strategy, we assume that documents are represented by classical bag-of-words. Given D^+ and D^-, the computation of the ideal document d_C^* is based on the Rocchio [4] algorithm as follows:

$$d_C^* = \frac{1}{|D^+|} \sum_{i \in D^+} d_i - \frac{1}{|D^-|} \sum_{j \in D^-} d_j \tag{4}$$

Moreover, we want to evaluate the performance of our approach when a reduced space, likewise a *Semantic Document Space*, is involved. The SDS is built by *Random Indexing* (RI) [1] a technique based on the Random Projection: the idea is that high dimensional vectors chosen randomly are "nearly orthogonal". This yields a result that is comparable to orthogonalization methods, such as Singular-Value Decomposition, but saving computational resources.

We set up a baseline system based on the BM25 multi-fields model [3].

The evaluation has been designed using the CLEF 2009 Ad-Hoc WSD Robust Task collection. To evaluate the performance we performed 150 runs by considering all possible combinations of the three parameters involved in our method: n (the cardinality of D^+), m (the cardinality of D^-) and the parameter α used for the linear combination of the scores (see Equation 1). We selected different ranges for each parameter: n ranges in $[1, 5, 10, 20, 40]$, m in $[0, 1, 5, 10, 20, 40]$, while α in $[0.3, 0.4, 0.5, 0.6, 0.7]$. The cardinality of D_{init} was set to 1,000.

Identifying relevant documents is quite straightforward: we assume the top ranked documents as relevant, while identifying non-relevant ones is not trivial. We proposed two strategies to select the set (D^-) of non-relevant documents, which are based on plausible heuristics rather than a theory:

1. *BOTTOM*, which selects the non-relevant documents from the bottom of the rank;
2. *RELJUD*, which relies on relevance judgements provided by CLEF organizers. This technique selects the top m ranked documents which are non-relevant exploiting the relevance judgements. We use this strategy to "simulate" the user's explicit feedback; in other words we assume that the user selects the first m non-relevant documents.

We evaluate each run in terms of MAP and GMAP over all the queries. Table 1 reports the results for the *baseline* and all three strategies (*Information Subtraction*, *VSM* and *SDS*). For each strategy, *positive* stands for the best run when only relevant documents were involved, while *BOTTOM* and *RELJUD* indicate the best run obtained for both strategies respectively. Improvements in percentage ($\Delta\%$) with respect to the baseline are reported.

The experimental results are very encouraging. Both methods (*BOTTOM* and *RELJUD*) show improvements with respect to the baseline in all the approaches. The main outcome is that quantum negation outperforms the "information subtraction" strategy.

Genarally, *BOTTOM* strategy results in not significant improvements, and in the case of "information subtraction", the introduction of non-relevant documents results in lower performance. The blind selection of non-relevant documents produces a side effect in "information subtraction" strategy due to the

Table 1. Evaluation results using all three strategies.

Method	*Run*	n	m	α	MAP	$\Delta\%$	$GMAP$	$\Delta\%$
-	baseline	-	-	-	0.4139	-	0.1846	-
Information Subtraction	positive	1	0	0.6	0.4208	+1.67	0.1754	-4.98
	BOTTOM	1	1	0.6	0.4175	+0.87	0.1750	-5.20
	RELJUD	40	40	0.7	0.5932	+43.32	0.2948	+59.70
Orthogonalization VSM	positive	1	0	0.5	0.4372	+5.63	0.1923	+4.17
	BOTTOM	1	5	0.6	0.4384	+5.92	0.1923	+4.17
	RELJUD	40	40	0.7	0.6649	+60.64	0.3240	+75.51
Orthogonalization SDS	positive	1	0	0.5	0.4362	+5.39	0.1931	+4.60
	BOTTOM	1	5	0.6	0.4367	+5.51	0.1928	+4.44
	RELJUD	40	40	0.7	0.6646	+60.57	0.3415	+84.99

information loss, while the quantum negation removes from relevant documents only those "negative" aspects that belong to the non-relevant ones.

As expected, the method $RELJUD$ obtains very high results. In this case quantum negation obtains very high improvements with respect to the "information subtraction" strategy. This proves that quantum negation is able to take advantage of information about non-relevant documents. The best results in $RELJUD$ are obtained when a lot of non-relevant documents are involved, but in a real scenario this is highly improbable. We performed several runs considering only one non-relevant document and varying the numbers of those relevant. The highest MAP value for SDS is 0.4606 (GMAP=0.2056), while for VSM is 0.4588 (GMAP=0.2028), both values are obtained with five relevant documents (these results are not reported for the sake of simplicity). Moreover, in both $BOTTOM$ and $RELJUD$ differences between SDS and VSM are not relevant.

These values support our thesis that negation expressed by quantum logic operator is able to model effectively the concept of non-relevance, opening new perspective for those tasks where the concept of non relevance plays a key role.

References

1. Kanerva, P.: Sparse Distributed Memory. MIT Press (1988)
2. Melucci, M., Rijsbergen, K.: Quantum mechanics and information retrieval. In: Melucci, M., Baeza-Yates, R. (eds.) Advanced Topics in Information Retrieval, Information Retrieval, vol. 33, pp. 125–155. Springer (2011)
3. Robertson, S., Zaragoza, H., Taylor, M.: Simple BM25 extension to multiple weighted fields. In: Proc. of the 13th ACM Int. Conf. on Information and Knowledge Management. pp. 42–49. ACM, New York, NY, USA (2004)
4. Salton, G., Buckley, C.: Improving retrieval performance by relevance feedback. Journal of the American Society for Information Science 41(4), 288–297 (1990)
5. Widdows, D., Peters, S.: Word vectors and quantum logic: Experiments with negation and disjunction. Mathematics of language (8), 141–154 (2003)

Hierarchical Text Classification for Supporting Educational Programs

Qi Ju∗, Chiara Ravagni⋆†, Alessandro Moschitti∗, and Giampiero Vaschetto⋆

∗DISI, University of Trento, Italy
{qi,moschitti}@disi.unitn.it

⋆Centro Studi Erickson, Italy
{chiara.ravagni,giampiero.vaschetto}@erickson.it

†University of Nuremberg, Germany

Abstract. More than two decades have passed since the first design of the CONSTRUE system [2], a powerful rule-based model for the categorization of Reuters news. Nowadays, statistical approaches are well assessed and they allow for an easy design of text classification (TC) systems. Additionally, the Web has emphasized the need of approaches for digesting large amount of textual information and making it more easily accessible, e.g., thorough hierarchical taxonomies like *Dmoz* or *Yahoo! categories*. Surprisingly, automated approaches have not proved yet to be indispensable for such categorization processes. This suggests that the role of TC might be different from simply routing documents to different topical categories.
In this paper, we provide evidence of the promising use of TC as a support for an interesting and high level human activity in the educational context. The latter refers to the selection and definition of educational programs tailored on specific needs of pupils, who sometime require particular attention and actions to solve their learning problems. TC in this context is exploited to automatically extract several aspects and properties from *learning objects*, i.e., didactic material, in terms of semantic labels. These can be used to organized the different pieces of material in specific didactic program, which can address specific deficiencies of pupils. The TC experiments, carried out with state-of-the-art algorithms and a small set of training data, show that automatic classifiers can easily derive labels like, *didactic context*, *school matter*, *pupil difficulties* and *educative solution type*.

Keywords: hierarchical text classification, information management applications, e-learning

1 Introduction

The last two decades have seen an impressive development of methods for automated text categorization (TC) [7]. This has been mainly due to the combination of two important factors: (i) the exponential development of the Web, requiring for effective methods of information access and management; and (ii) the enhancement in theory and practice of machine learning methods, which constitute the bases of TC.

Despite the success of the TC research, it is still not clear if such technology should be devoted to the design of topical categorization systems as very famous Web hierarchical categorization systems are currently manually maintained, e.g., *Dmoz* or *Yahoo! categories*. On the other hand, TC also regards the association of semantic labels that go beyond the simple routing of information to the most appropriate user feeds. Indeed, this kind of task inevitably suffers from errors in Recall and/or in Precision. Different would be the approach and results, if the outcome of the TC system were cooperatively used as a tool to organize the information in different and creative ways. In this respect, TC would be seen as a tool similarly

to search engines, rather than an end-to-end system forced to demonstrate a very high accuracy.

In this paper, we report on our experience with the e-Value project, whose aims are the reorganization or combination of educational materials in different pedagogical contexts. The Erickson Research Centre has been cataloging a large set of published educational materials in smaller units, according to the SCORM (2004) standards, Shareable Content Object Reference Model[1]. These documents are used for the creation of novel and specific didactic product as follows: (i) school classes are evaluated about target cognitive processes; (ii) processes in which pupils have difficulties are detected and recorded in a huge database (DB) of normative data along with the results of its elaboration; (iii) The Decision Support System (DSS) chooses the proper didactic material for the class according to the DB content.

The above steps require: (a) to identify cognitive processes involved in pupils' learning; (b) to divide the didactic materials in smaller parts (learning objects); and (c) classify such objects according to their bibliographic characteristics and to the cognitive processes involved, which depends on the user context (e.g., age, class, special situations). An automatic classifier can be used for easing and speeding up the last step. It can provide a rough classification, which can constitute the starting point for the work of expert catalogers.

The use of the classifier would reduce the cataloging costs, both in terms of time and human resources. Indeed, any educational material, being part of a book, article or best practice, needs to be read and evaluated by experts, before being assigned to the proper categories; this process takes a huge amount of time. As an alternative model, the classifier can perform a first approximate categorization and after, the experts can refine it. The clear advantage is that materials pertaining to a certain subject can be directly assigned to its experts (working in that field), thus improving the accuracy of classification and avoiding the burden to exchange materials among the different experts.

However, the above scenario could be realized only if the adopted multi-class classifier (MCC) performed accurate hierarchical categorization. Given the novelty of the intended taxonomy, it is not simple to predict if MCC can deploy the needed accuracy. For this purpose, we have:

- designed a new taxonomy that meets the organization needs of e-Value;
- defined an annotation procedure and produced an initial datasets of 122 documents, organized in 112 categories (of course the documents are repeated in the hierarchy); and
- implemented an MCC, which exploits state-of-the-art TC models such as, Support Vector Machines, structured in binary flat categorizers.

The preliminary experiments on the overall hierarchy of 112 nodes show promising results, ranging from a Micro-F1 of above 95% for the first level to about 70% on the whole hierarchy. This outcome is rather promising and enables future research in the use of TC for the efficient implementation of educational programs.

In the reminder of this paper, Section 2 describes the tackled task in more detail, Section 3 reports on our results and Section 4 derives the final conclusions.

2 Automatic Support to the e-Value project

The main objective of the e-Value project is to design, develop and test a multimedia platform (consisting of a set of web applications), which integrates the evaluation of various learning abilities and the application of didactic processes. These can benefit from automatic methods for classifying the didactic material used in such

[1] `http://www.adlnet.gov/capabilities/scorm/scorm-2004-4th`

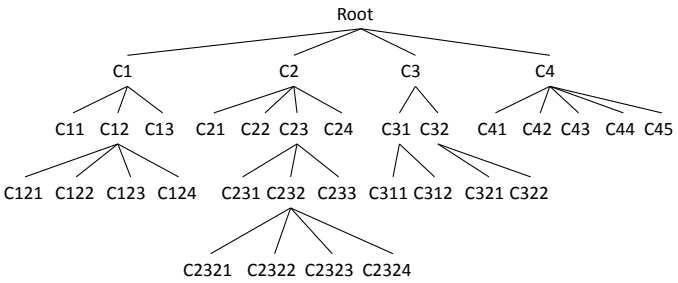


Fig. 1. Hierarchical Categorization Scheme of e-Value (only category with at least 1 training documents are present

processes. The next sections describe the problem in more detail and suggest how a TC system can be used in such context.

2.1 e-Value Framework

The framework includes different interconnected processes:

- standard evaluation procedures and dynamic assessment of learning abilities of pupils;
- collection of normative data, e.g., educational material and pupils' evaluations;
- continuous data flow, i.e., the related database is continuously updated and the normative data currently available is integrated and compared with the new arriving data; and
- qualitative and quantitative evaluation of the collected data.

The educational material is used for defining didactic products, which address specific action (intervention). It consists of books, CD-ROMs, collections of articles, etc. The e-Value project aims at both using independently and jointly the materials above.

Designing an intervention often requires the use of units taken from several books or CD-ROMs but including the entire sources is very ineffective, considering that only some small parts will be used. To enable more flexibility in the creation of training programs, the material collections are divided into basic training units, called learning objects, which can be reassembled in a flexible way. This requires to analyze the materials to be used in the interventions and selecting the portion involved in the target cognitive processes.

2.2 A framework use-case

A use of the framework is illustrated by the following example. In a school context some classes are evaluated with respect to targeted cognitive processes. The tests may reveal that some of the pupils have difficulties in certain processes. Thus, the test results are recorded (building a large database of normative data) along with some elaboration of them, i.e., basic data statistics. Then the DSS chooses the proper didactic material for the class by proposing different material to pupils requiring attention and quick intervention. For this purpose the educational team need to:

- identify every cognitive process that can be involved in learning. At the moment, this has been restricted to mathematics and reading-writing (with linguistic skills and metaphonetics);

C1	**Categorizzazione contesto didattico**	C22	Metafonologia	C2435	Espressioni
C11	Scuola dell'infanzia	C221	Globale	C2436	Potenze
C12	Scuola primaria	C2211	Rima	C2437	Radici quadrate
C121	Primaria Classe I	C2212	Sillaba	C244	Calcolo-Numeri razionali
C122	Primaria Classe II	C222	Profonda	C245	Calcolo-Numeri relativi
C123	Primaria Classe III	C2221	Fonema	C246	Calcolo-Rapporti e proporzioni
C124	Primaria Classe IV	C23	Abilità linguistiche	C247	Calcolo-Calcolo letterale
C125	Primaria Classe V	C231	Lessico	C248	Problem-solving
C13	Scuola secondaria di 1 grado	C2311	Denominazione	C249	Capacità di orientarsi nello spazio
C2	**Categorizzazione materia**	C2312	Categorizzazione	C24_10	Costruire sistemi di riferimento convenzionali
C21	Letto-scrittura	C2313	Identificazione	C24_11	Geometria euclidea (piana)
C211	Prerequisiti	C2314	Definizione	C24_12	Misura di grandezze geometriche
C2111	Prerequisiti grafo motori	C2315	Polisemia	C24_13	Misura di grandezze fisiche
C2112	Prerequisiti visuo spaziali	C2316	Arricchimento lessicale	C24_14	Le trasformazioni geometriche
C2113	Teorie ingenue	C232	Morfo-sintassi	C25	Altro
C212	Decodifica	C2321	Concordanze	C3	**Categorizzazione situazione alunni**
C2121	Lettere	C2322	Struttura della frase	C31	BES
C2122	Sillabe	C2323	Analisi grammaticale	C311	Autismo
C2123	Parole	C2324	Analisi logica	C312	Udito
C2124	Non parole	C233	Narrazione	C313	Vista
C2125	Frasi-Brano	C2331	Comprensione racconto	C314	Psicomotricità
C213	Comprensione	C2332	Produzione racconto	C315	Sindrome di Down
C2131	Parole	C24	Matematica	C316	Altro
C2132	Frasi	C241	Numero	C32	DSA
C2133	Brano	C2411	Processi semantici	C321	Iperattività
C214	Compitazione	C2412	Conteggio	C322	Dislessia
C2141	Clettere	C2413	Processi pre-sintattici	C323	Disgrafia
C2142	Sillabe non ortografiche	C2414	Processi lessicali e sintattici	C324	Discalculia
C2143	Parole non ortografiche	C242	Calcolo - processi di base	C325	Combinazione di DSA diversi - altro
C2144	CNon parole	C2421	Segni delle operazioni	C326	Nessun DSA
C215	Ortografia	C2422	Fatti numerici	C4	**Categorizzazione tipo di intervento**
C2151	Oparole	C2423	Tabelline	C41	Potenziamento
C2152	Ofrasi	C2424	Calcolo a mente	C42	Recupero
C2153	Obrano	C243	Calcolo - numeri naturali	C43	Didattica insegnamento
C216	Stesura testo	C2431	Algoritmi di calcolo scritto	C44	Intervento logopedico
C2161	Pianificazione	C2432	Incolonnamento di numeri	C45	Intervento psicologico
C2162	Trascrizione	C2433	Multipli e divisori		
C2163	Revisione	C2434	Minimo Comune Multiplo e Massimo Comune Denominatore		

Table 1. Description of the different categories of the hierarchy in Figure 1.

Level_1	Train_No	Test_No	Precision	Recall	F1
C1	38	16	0.8421	1.0000	0.9143
C2	40	20	0.9048	0.9500	0.9268
C3	41	19	0.9500	1.0000	0.9744
C4	39	17	1.0000	0.8824	0.9375
Micro			0.9200	0.9583	0.9388
Macro			0.9242	0.9551	0.9382

Fig. 2. Performance for the first level

- divide the didactic materials in smaller parts (learning objects). This because the use of the entire books or CD-Rom would be unfeasible, considering that just a few exercises need to be applied. Thus the whole material has to be checked by experts to be subdivided in learning objects. The latter are then used to design the formative offer, in place of the entire material, obtaining a more personalized and individualized learning.
- Categorize the materials according to their bibliographic characteristics and, most importantly for the fruition of the materials, to features of the involved cognitive processes, e.g., the age, class and special situations of the target pupils etc.
- Porting the material from paper or optical media to an electronic format (pdf or swf) so that it can be reassembled online and offline.

In the last phase the application of an automatic classifier can provide significant benefits to the whole process as explained in the following section.

2.3 Classification Task

To meet the need of the e-Value project, we have defined a new taxonomy as well as the annotation procedure and initial datasets. Our hierarchical categorization scheme is shown in Figure 1, whose more descriptive labels are reported in Table 1. The materials have to be classified according to four macro-categories, and then divided into a structure of sub-categories of 4 levels. Each category is meaningful for a correct description of the materials, from both administrative perspective (e.g., in which educational context should be applied) and subject/cognitive process viewpoint (e.g. Mathematics – Number – Lexical and semantic processes instead of Mathematics – Basic processes of calculus – Numerical facts). The Macro-categories are: C1 – School and class (referring to the ages 5 – 14); C2 – Subject/cognitive process (referring to the subjects of mathematics, linguistics, phonetics, reading-writing abilities); C3 – Pupils' situation (for the cases of special needs or particular situations); and C4 – Type of material (or the normal didactic usage in the class, or for pupils with special situation or greater difficulties in the subject).

Such automatic classification could improve the manual categorization costs, in terms of both time and human resource. Each piece of educational material, being part of a book, article or best practice, needs to be read and evaluated by experts, before being assigned to the proper categories, and this process takes a huge amount of time. Therefore, the use of an automatic classifier could significantly reduce the time required to read and evaluate the materials. Of course, experts will need to read part of the material in any case to refine and validate the output of the classifier. However, the materials pertaining to a certain subject can be directly routed to the experts of such field, thus improving the categorization accuracy.

Level_2	Train_No	Test_No	Precision	Recall	F1
C1	38	16	0.8421	1	0.9143
C2	40	20	0.9048	0.95	0.9268
C3	41	19	0.95	1	0.9744
C4	39	17	1	0.8824	0.9375
C11	5	1	0	0	0
C12	36	15	0.9333	0.9333	0.9333
C13	7	1	0	0	0
C21	12	5	1	0.8	0.8889
C22	10	3	0.4	0.6667	0.5
C23	4	1	0.9412	0.9412	0.9412
C24	20	11	1	1	1
C25	0	1	0	0	0
C31	2	0			
C32	39	19	0.95	1	0.9744
C41	23	11	1	0.9091	0.9524
C42	31	12	0.8571	1	0.9231
C43	25	8	0.8889	1	0.9412
C44	10	6	1	0.6667	0.8
C45	0	1	0	0	0
Micro			0.9162	0.9162	0.9162
Macro			0.6408	0.637	0.6325

Fig. 3. Performance for the second level

3 Experiments

The aim of our evaluation is to demonstrate that state-of-the-art TC methods can be applied to learn hierarchical classifiers for our e-Value taxonomy. This task is made complex by two different aspects: (i) in addition to topic labels such as, *Euclidean Geometry*, *Problem Solving* or *Geometric Transformation*, the taxonomy also contains semantic characterization such as *Story Development* or *Story Understanding*, whose characterization using simple terms seems harder; and (ii) given the novelty of the taxonomy, we could only produce a small dataset, which makes the learning of classification functions more difficult. To deal with and analyze such problems, we experimented with hierarchy subsets, defined according to the hierarchy's levels, ranging from 1 to 4 (the maximum depth of our hierarchy). The deeper the level, the more difficult TC is.

3.1 Setup

One major drawback of machine learning and thus of TC based on it is the need of training data, i.e., a set of documents manually classified into the referring taxonomy. This data is difficult to find and/or to produce as it requires human labor. Given the novelty of our taxonomy defined in Figure 1, no previous data was available. Thus, we set an annotation procedure (with only one annotator) of the didactic material available in the Erickson's database. We randomly selected 60 documents and we classified each of them according to all the 112 nodes of the taxonomy. This led to a dataset of 122 documents (repetitions are considered).

We randomly divided the above data in training and test set by taking care that for each document all its repetitions were all put either in the training or in the test set. The training data was used to learn the set of 112 binary classifiers, one for each category, following the one-vs-all schema. The output of the multi-class classifier is the merged set of the individual binary classifier decisions. Although simple, this is considered a state-of-the-art approach [5, 3]. We used default SVM parameters as the small training data prevented to apply any reasonable parameterization approach. We used a bag-of-term representation (string separated by space and punctuation) without applying any feature selection, stop list or lemmatization. Although, we are

Level_3	Train_No	Test_No	Precision	Recall	F1
C121	23	10	0.6667	0.6	0.6316
C122	20	8	0	0	0
C123	10	8	0	0	0
C124	10	5	1	0.6	0.75
C125	9	3	0.3333	0.3333	0.3333
C212	5	1	0	0	0
C214	0	1	0	0	0
C215	0	2	0	0	0
C216	2	1	0	0	0
C221	9	2	0.4	1	0.5714
C222	9	3	0.6667	0.6667	0.6667
C232	2	1	0	0	0
C241	1	2	0	0	0
C242	12	10	1	0.4	0.5714
C243	1	4	0	0	0
C321	0	3	0	0	0
C322	20	9	0.7	0.7778	0.7368
C323	1	6	0	0	0
C324	16	4	1	1	1
C325	5	1	0	0	0
Micro			0.8545	0.7251	0.7845
Macro			0.2883	0.2689	0.2631

(a) third level

Level_4	Train	Test	Precision	Recall	F1
C2121	2	1	0	0	0
C2122	3	1	0	0	0
C2123	3	1	0	0	0
C2142	0	1	0	0	0
C2151	0	2	0	0	0
C2152	0	1	0	0	0
C2153	0	1	0	0	0
C2161	2	1	0	0	0
C2211	4	1	1	1	1
C2212	9	2	0.4	1	0.5714
C2221	9	3	0.6667	0.6667	0.6667
C2322	0	1	0	0	0
C2323	2	1	0	0	0
C2324	1	1	0	0	0
C2411	1	2	0	0	0
C2421	5	3	0	0	0
C2422	4	5	0	0	0
C2423	1	2	0	0	0
C2424	3	6	1	0.1667	0.2857
C2431	1	4	0	0	0
C2432	1	1	0	0	0
C2433	0	2	0	0	0
Micro			0.8430	0.6395	0.7273
Macro			0.1394	0.1288	0.1147

(b) fourth level

Fig. 4. Performance for the third and fourth level. Categories with no document in the test set and the categories of upper levels are not reported.

confident that the latter may relevantly improves our models. We used the classical $log(TF) * IDF$ weighting scheme and normalized vectors.

The performance is provided by means of Micro- and Macro-Average F1, evaluated from our test data over all 112 categories. Additionally, the F1s of the binary classifiers are reported. For measuring the performance of different hierarchical levels, only the nodes up to the target level are considered, e.g., for the first level, we only measure the Micro/Macro F1 of C1, C2, C3 and C4.

3.2 Results and Discussion

Table 2 reports the performance on the first level. We note that for each category there are about 40 documents for training. These seem to be enough as the accuracy of the individual categories as well as the overall Micro/Macro F1 is exceptionally high. This is not completely surprising as most documents are repeated in the above four categories.

Table 3 illustrates the results for the second level. We note that when the training documents are more than 20, very good results can be achieved. Low performance is shown for C11 and C13, which are trained with less than 7 documents. Additionally, they have only one test document, this means that their accuracy cannot really be estimated. The situation of C31 is even worse as it has no test documents. In this case, we do not report any accuracy in the related row. It should also be noted that, since we use one-vs-all schema, the accuracy of C1,..,C4 is the same as before. Thus, from now on, we will not report the accuracy of previously reported binary classifiers.

Table 4 shows the performance on levels 3 and 4. Again the few training documents available for the classifiers prevent to achieve a reasonable F1. There are some good cases such as C124 and C322 but also bad cases such as C122 and C123. The latter two refer to *Primaria Classe II* and *Primaria Classe III*, respectively,

which have large overlap with the other classes, i.e., I, IV and V. For separating such categories, the simple bag-of-words may not be enough.

4 Conclusions

In this paper, we have described an interesting and new semantic classification problem in the context of the educational framework of the e-Value project. We have defined a new hierarchical taxonomy, which is promising for improving the production cycle of educational systems. To test the feasibility of the approach, we have also built a corpus annotated according to the above taxonomy. Such data was used for training an MCC based on SVMs. The results show that when there is a reasonable amount of training documents the classifiers can deploy remarkably high accuracy. On the other hand, the F1 of lower level categories is highly affected by data scarceness. Some categories would probably require the definition of more expressive features to better model their separation.

Possible solutions are also provided by previous work, which shows more advanced TC models, e.g., [6], in which global dependencies between hierarchical nodes are encoded in a gradient descendent learning approach. They experimented with Reuters Volume 1 (RCV1) [2] on a subhierarchy only containing 34 nodes. Other relevant work such as [4] and [1] uses a rather different datasets and a different idea of dependencies based on the feature distributions over the linked categories. Finally, [3] experiment with models similar to ours achieving state-of-the-art on RCV1.

Acknowledgements

The research described in this paper has been partially supported by the Italian Project e-Value (PAT) and by the European Community's Seventh Framework Programme (FP7/2007-2013) under the grants #231126: LivingKnowledge – Facts, Opinions and Bias in Time, #247758: EternalS – Trustworthy Eternal Systems via Evolving Software, Data and Knowledge, and #288024: LiMoSINe – Linguistically Motivated Semantic aggregation engiNes.

References

1. Dumais, S.T., Chen, H.: Hierarchical classification of web content. In: Belkin, N.J., Ingwersen, P., Leong, M.K. (eds.) Proceedings of SIGIR-00, 23rd ACM International Conference on Research and Development in Information Retrieval. pp. 256–263. ACM Press, New York, US, Athens, GR (2000), http://research.microsoft.com/~sdumais/sigir00.pdf
2. Hayes, P.J., Weinstein, S.P.: Construe/Tis: a system for content-based indexing of a database of news stories. In: Rappaport, A., Smith, R. (eds.) Proceedings of IAAI-90, 2nd Conference on Innovative Applications of Artificial Intelligence. pp. 49–66. AAAI Press, Menlo Park, US (1990)
3. Lewis, D.D., Yang, Y., Rose, T., Li, F.: Rcv1: A new benchmark collection for text categorization research. The Journal of Machine Learning Research (5), 361–397 (2004)
4. McCallum, A., Rosenfeld, R., Mitchell, T.M., Ng, A.Y.: Improving text classification by shrinkage in a hierarchy of classes. In: ICML. pp. 359–367 (1998)
5. Rifkin, R., Klautau, A.: In defense of one-vs-all classification. J. Mach. Learn. Res. 5, 101–141 (December 2004), http://dl.acm.org/citation.cfm?id=1005332.1005336
6. Rousu, J., Saunders, C., Szedmak, S., Shawe-Taylor, J.: Kernel-based learning of hierarchical multilabel classification models. The Journal of Machine Learning Research (7), 1601–1626 (2006)
7. Sebastiani, F.: Machine learning in automated text categorization. ACM Computing Surveys 34(1), 1–47 (2002)

[2] trec.nist.gov/data/reuters/reuters.html

Error-Correcting Output Codes for Multi-Label Text Categorization

Giuliano Armano[1], Camelia Chira[2], and Nima Hatami[1]

[1] Department of Electrical and Electronic Engineering
University of Cagliari
Piazza D'Armi, I-09123 Cagliari, Italy
[2] Department of Computer Science
Babes-Bolyai University
Kogalniceanu 1, Cluj-Napoca 400084, Romania

Abstract. When a sample belongs to more than one label from a set of available classes, the classification problem (known as multi-label classification) turns to be more complicated. Text data, widely available nowadays in the world wide web, is an obvious instance example of such a task. This paper presents a new method for multi-label text categorization created by modifying the Error-Correcting Output Coding (ECOC) technique. Using a set of binary complimentary classifiers, ECOC has proven to be efficient for multi-class problems. The proposed method, called ML-ECOC, is a first attempt to extend the ECOC algorithm to handle multi-label tasks. Experimental results on the Reuters benchmarks (RCV1-v2) demonstrate the potential of the proposed method on multi-label text categorization.

Keywords: Ensemble learning, Error-Correcting Output Coding (ECOC), Information filtering and retrieval, Multi-label Classification, Multi-label Text Categorization (ML-TC).

1 Introduction

Text Categorization (TC), also known as document classification, plays a key role in many information retrieval (IR) -based systems and natural language processing (NLP) applications. First research on TC goes back to Maron's [1] seminal work on probabilistic text classification. Since then, TC has been used for a number of different applications using techniques from machine learning, pattern recognition and statistics. In [3], TC applications are grouped into hierarchical categorization of web pages, word sense disambiguation, automatic indexing for boolean IR systems, document filtering and organization. Speech categorization as combination of a speech recognition and TC methods, multimedia document categorization through the analysis of textual captions, author identification for literary texts of unknown or disputed authorship, language identification for texts of unknown language, automated identification of text genre, and automated essay grading are some examples for such applications in real-world problems [4, 6] .

The traditional classification problem in pattern recognition refers to assigning any incoming sample to one of two (binary problem) or more (multi-class problem) distinct predefined classes. An even more complex scenario - called multi-label classification - is one in which the classes have overlap between each other. TC or automatically labeling natural language texts with thematic categories from a predefined set is one such task. An instance document or web page about "Persian carpet exhibition" can belong to both "economy" and "art" categories. Despite its multi-label nature, the majority of research studies on TC have considered it as single-label task by assigning the samples into only one of the existing classes. However, this approach simplifies the task and handles it using a huge bibliography of learning algorithms, yet failing to provide a complete solution to multi-label TC.

There are two main approaches in the literature to deal with multi-label classification: *(i) Problem transformation* approaches which transform the multi-label problem into one or more single-label problems, and *(ii) Algorithm adaptation* approaches which extend specific learning algorithms in order to handle the multi-label task directly. Although many approaches have been proposed based on different kinds of classifiers and architectures over a variety of application domains, there is no clear winner method over the rest (see [21] [22] for some recent surveys) and each of them has its own advantages and disadvantages.

Classifier ensembles (also known as Multiple Classifier Systems) is a paradigm based on the *divide-and-conquer* strategy to deal with complex classification problems. The main idea is to use an ensemble of simple *base-classifiers*, each applied to a sub-task, instead of hiring a single classifier expected to take care of the entire task. This strategy typically improves a classification system in terms of stability and classification accuracy (bias-variance reduction). Bootstrap aggregating (i.e., bagging) is a machine learning technique that combines a number of base-classifiers, each trained on a set of bootstrap samples of the original data [16]. The boosting strategy is a fixed point procedure aimed at iteratively generating a set of *week learners* [17]. Random Subspace Ensemble (RSE) [18] creates a set of base classifiers, each using only a (randomly determined) subset of the original feature space. RSE is particularly effective for high-dimensional classification problems. The Mixture of Experts (ME) [13] stochastically partitions the input space of the problem into a number of subspaces, so that experts become specialized on each subspace. The ME uses another expert called *gating network* to manage this process - which is trained together with the experts. Finally, Error-Correcting Output Codes (ECOC) [14] is an ensemble making strategy inspired by the coding theory which decomposes any multi-class problem into some complementary binary sub-problems using a (normally pre-defined) codematrix. The final multi-class solution is obtained by aggregating the binary outputs.

This paper proposes a method for multi-label TC called ML-ECOC created by extending the ECOC strategy. ML-ECOC modifies the coding/decoding phases of the standard ECOC algorithm making it suitable to the multi-label problems. This modification includes setting up new rules in both coding and decoding phases to avoid the occurrence of any inconsistency while handling

multi-label data. Experiments on the text mining problem of Multi-Label Text Categorization (ML-TC) show a good performance of the proposed ML-ECOC. Comparissons to the state-of-the-art methods from different perspectives are carried out and the obtained results are analysed in detail.

The rest of this paper is organized as follows: the standard ECOC algorithm presented in section 2, the proposed ML-ECOC algorithm is presented in section 3 with full details, section 4 presents the analysis of experimental results on Reuter's version 2 datasets and the comparisons with the state-of-the-art methods from litrature. Last section concludes the paper and discusses some directions of future work.

2 Error-Correcting Output Coding

ECOC is a classifier ensemble method inspired by signal transmission in information theory used to safely send and receive the data. Besides its *error-correcting* capability to recover the errors made in each sub-problem classification level, ECOC has the advantage of decomposing a multi-class problem into some binary sub-problems (*dichotomies*) in machine learning concept. Each sub-problem is tackled by a *dichotomizer* and the final solution for the multi-class problem is created by aggregating the results of the dichotomizers (divide-and-conquer principle). For this reason, ECOC performs well particularly on the problems with large number of classes for which other classifiers normally have difficulties.

Given a classification problem with N_c classes, the main idea of ECOC is to create a binary/ternary *codeword* for each class. Arranging the codewords as rows of a matrix, we define a *codematrix* M, where $M \in \{-1, 0, +1\}^{N_c \times L}$ and L is the code length (coding phase). From a learning point of view, M specifies N_c classes to train L dichotomizers, $f_1 ... f_L$. A classifier f_l is trained according to the column $M(., l)$. If $M(i, l) = +1$ then all examples of class i are positive, if $M(i, l) = -1$ then all its examples are negative *supper-class* and, finally, if $M(i, l) = 0$ none of the examples of class i participate in the training of f_l.

Let $\overline{y} = [y_1 ... y_L]$, $y_l \in \{-1, +1\}$ be the output vector of the L classifiers in the ensemble for a given input x. In the *decoding* phase, the class output that maximizes the similarity measure s (e.g. the Hamming distance) between $\overline{y}$ and row $M(j, .)$ (its codeword) is selected:

$$Class\ Label = ArgMax\ S(\overline{y}, M(j, .)) \tag{1}$$

The ECOC matrix codifies the class labels in order to achieve different partitions of classes, considered by each dichotomizer. The main coding strategies can be divided into problem-independent (or fixed) and problem-dependent. Most popular pre-designed problem-independent codeword constructions satisfy the requirement of high separability between rows and columns in order to increase error-correcting capability and diversity between dichotomies. These strategies include: *1vsA*, using N_c dichotomizers, each trained to discriminate a given class

from the rest of classes; *random techniques*, which can be divided into the *dense-random*, consisting of a binary matrix with high distance between rows with estimated length of $10 \log_2 N_c$ bits per code, and the *sparse-random* strategy based on the ternary symbol and with the estimated length of about $15 \log_2 N_c$. *1vs1* is one of the most well-known coding strategies, with $N_c(N_c - 1)/2$ dichotomizers including all combinations of pairs of classes [12]. Finally, BCH codes [2] are based on algebraic techniques from Galois Field theory and, while its implementation is fairly complex, it has some advantages such as generating ECOC codewords separated by a minimum, configurable Hamming distance and good scalability to hundreds or thousands of categories. Moreover, recently some researchers [10, 9, 11] argue that, unlike the problem-independent strategies where a codematrix is defined without considering the problem characteristics or the classification performance, the selection and the number of dichotomizers must depend on the performance of the ensemble for the problem at hand.

3 Multi-Label ECOC for TC

The first application of ECOC algorithm on TC dates back to 1999 [8, 7], However, in these studies, the authors simply use standard single-label classifiers and view the problem as a traditional multi-class classification. Since then, many researchers also used ECOC with different types of classifiers on various applications but with more or less the same assumptions. From the ECOC literature, one can conclude that there are three main possible ways to improve ECOC classifiers: (i) code matrix design, (ii) building binary classifiers, and (iii) decoding step. In TC area, the improvements are mainly limited to the second option i.e. building binary classifiers as accurate as possible. This goal is achieved in [20] by Model-Refinement strategy which is used to adjust the so-called bias in centroid classifiers. The basic idea is to take advantage of misclassified examples in the training data to iteratively refine and adjust the centroids of text data. In [19], Li et al. proposed a simple strategy to improve binary text classification via multi-class categorization (dubbed 2vM) for applications where sub-class partitions of positive and/or negative classes are available. As multi-class categorization may implicitly capture the interactions between sub-classes, detailed subclasses are expected to help differentiating the positive and negative classes with high accuracy.

The reason that all these works are limited to single-label assumption is that an *inconsistency* would occur otherwise in ECOC classification while applying to multi-label data. For instance, imagine a document d belongs to a label set $[1, 3, 5]$, each label representing a content based topic. Also imagine 5-th column of an instance (predefined or given) matrix $M^{7\times 9}$ shown in Figure 1 which is used to create dichotomizer f_5. Considering $d \rightarrow \omega = [c_1, c_3, c_5]$, now the question is which super-class sample d belongs to (+1 or -1)? According to traditional decoding of ECOC, the sample belongs to both super-classes of the dichotomy at the same time. This inconsistensy in assignment of d is not only limited to f_5 but also occures for dichotomies 3,4, 6, 7 and 8. In fact, standard ECOC algorithm

is only capable of single-label prediction for a traditional multi-class problem while it suffers from lack of capability to handle multi-label data in general. Therefore, a modification in the ECOC algorithm is required such that it can directly address multi-label data in both training the dichotomizers and label set prediction without any assumption and limitation. As mentioned before, the only way to address this issue so far was simplifying the problem to single-label classification [7, 8].

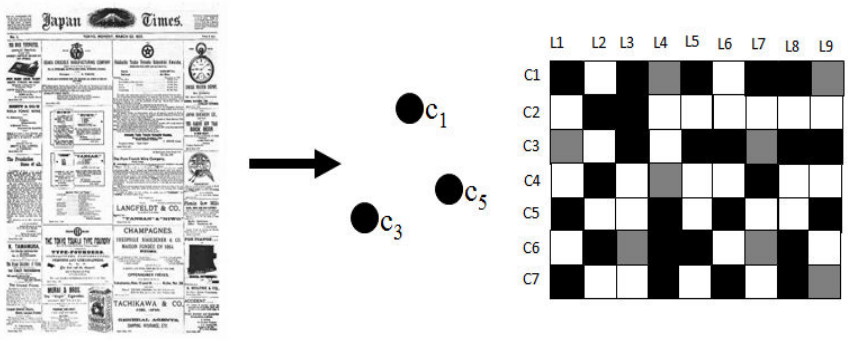


Fig. 1. An instant document d belogs to classes 1, 3 and 5 defined based on its content (Left). An instant codematrix with 7 rows (for 7 class-nodes) and 9 columns. Black, gray and white boxes represent -1, 0 and +1, respectively (Right).

Although the single-label assumption may be true in some TC applications, it certainly limits the application of ECOC to real-world multi-label cases. This is the point where ECOC algorithm requires a major modification to be applicable to multi-label problems. In the following, we introduce the ML-ECOC method to address any multi-label problem without any constraint and restricting assumption.

The main idea of ML-ECOC is to generate a codeword for each category of a TC task with only +1 (positive class) and 0 (don't care) bits. Unlike standard ECOC algorithm, where at least one +1 and one −1 bits are required at each column to define a dichotomy, to be non-zero is all ML-ECOC needs for a column. A classifier defined according to each column of the ML matrix and used to calculate degree of membership of d into a super-class which includes one or more categories. The inconsistency in the dichotomizing process is avoided by defining only positive class and *neutral* set which can not have any overlapping area. It is worth noting that a document belongs to ith positive class *if and only if* at least one of its labels from the label set is in the ith super-class. A document d (Figure 2) either should belong to positive class of ith column or its neutral set. For instance, d is a member of $2, 3, 4, 5, 6, 7$ and 8 positive class sets while should be considered as neutral for 1st and 9th.

Subsequently, it is obvious that this modification requires also different decoding strategy, since standard Euclidean or Hamming distances with $ArgMax$ labeling are not applicable anymore. Let us suppose a predicted codeword $\overline{y}_d = [\overline{y_1}...\overline{y_L}]$, $0 \leq \overline{y_l} \leq +1\}$ is a string assigned to document d (each bit representing the output of a classifier i.e. $\mathcal{P}_l(+1 \mid d)$). The posterior probability of each class using ML-ECOC is calculated as follows:

$$\mathcal{P}(c_N \mid d) = \frac{1}{\mid M(N,.) \mid} \sum_{l=1}^{L} \mathcal{P}_l(+1 \mid d) M(N,l) \tag{2}$$

For each document, ML-ECOC sorts categories by score and assigns YES to each of the t top-ranking categories. Parameter t is an integer ranging from 1 to the number of categories N_c whose value can be either specified by the user or automatically tuned using a validation set. It should be noted that when $t = 1$, this multi-label assignment turns into the standard single-label TC with $ArgMax$ rule. Obviously, it is just typical thresholding strategy adopted to ML-ECOC and the other existing throsholding methods can be applied. The generic ML-ECOC is summarized in Algorithm 1.

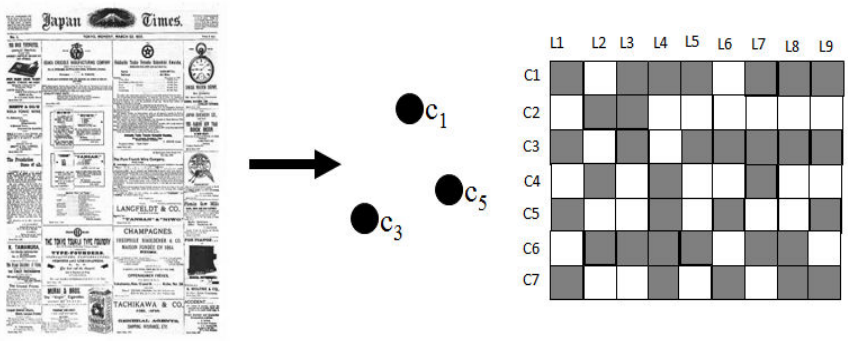


Fig. 2. ML-ECOC defines a binary codeword for each category of TC and sets up the decoding rule such that the problem decomposed in some subsets in which a positive supper-class stands against a neutral set. The number of columns, L varies depend on the coding method. Gray and white boxes represent 0 and +1 which represent positive and neutral data, respectively.

3.1 Why does ML-ECOC work?

The success of the ML-ECOC idea can be attributed to following three factors:

1. Unlike the standard TC approaches trying directly to discriminate different classes, ML-ECOC transfers the entire class space to many super-classes, which are not necessarily carrying meaningful concepts, by mixing them . This is helpful particularly to deal with what is called in the literature *Data sparsity*. This

Algorithm 1 ML-ECOC.

Input: $\mathcal{X}_t$, $\mathcal{T}_t$ training set, $\mathcal{X}_e$, $\mathcal{T}_e$ testing set and f learning algorithm.

Training:
- generate a binary codemtrix $M^{N_c \times L}$ which N_c is the number of categories and L varies with coding strategy.
- for i-th column in M:
  build (create) one-class set made of $\mathcal{T}_i^+$ and $\mathcal{T}_i^*$ supper-classes (positive and neutral sets respectively)
  train i-th classifier f_i with i-th training set

Testing:
- apply $\mathcal{X}_e$ on entire set of f_is
- create a codeword which i-th bit is $f_i(\mathcal{X}_e) = \mathcal{P}_i(+1 \mid \mathcal{X}_e)$
- calculate the posterior probability for each class using Eq. 2

- use multi-label decoding to predict label set

Output: $\bar{\omega} = [\bar{c_p}, \bar{c}_q, \bar{c}_r]$

is a measure for how much data we have for a particular dimension/entity of the model. A dataset is sparse if the number of samples for each class is not enough for a classifier to discriminate it from the rest which is normally the case in the TC problem. Therefore, mixing categories by ML-ECOC decomposing, not only used to define new class-boundaries which might provide additional information in final decision making, but also provides new one-class problems with more samples per positive class (in the case each super-class has more than one category). For instance, each super-class in first dichotomy of Figure 2 is made of 3 categories.

2. No matter which TC approach is chosen, a class-label is assigned to a document if its corresponding classifier *fires*. In fact, when a category is wrongly detected, there is no any *efficient* way to go back and fix it without the increase of the algorithm complexity and computational cost. However, in ML-ECOC there is no *dedicated* classifier for each category and decisions are made by *consensus* of all classifiers. Therefore, because of its *error-correcting* capability, even if some errors occur in the bit level, the final decision can still be reliable.

3. Another important issue arising while dealing with TC refers to *class-imbalanced* datasets where there is no balance between the positive and negative set of a category. This problem can badly affect the learning process particularly in the *Local Classifier per Category* approach when a category stands against the rest. ML-ECOC keeps more balance between two resulted positive classes and neutrals by having chance of including more than one class in the positive class set. For instance *Sparse-random* method can possibly include more than one category in a positive class resulting into more balanced data. Consequently,

efficient learning of the class boundaries by classifiers results in more accurate prediction.

4 Numerical Experiments and Results

For the text categorization experiments, we have chosen two commonly used multi-label datasets i.e. the Reuters (RCV1-V2) and TMC2007. A brief description of each is given below.

RCV1-V2: Reuters Corpus Volume1-Version2 is a large-scale dataset for text classification task. It is based on the well known benchmark dataset for text classification, the Reuters (RCV1) dataset. We use the topics full set 3 that contains (804,414) news articles. Each article is assigned to a subset of the 103 topics. A detailed description of the RCV1 dataset can be found in [5]. We preprocessed RCv1v2 documents as proposed by Lewis et al. [5] and, in addition, we separated the training set and the testing set using the same split adopted in [5]. In particular, documents published from August 20, 1996 to August 31, 1996 (document IDs 2286 to 26150) are included in the training set, while documents published from September 1, 1996 to August 19, 1997 (document IDs 26151 to 810596) are considered for testing. The result is a split of the 804,414 documents into 23,149 training documents and 781,265 test documents. In order to save computational resources, we have randomly chosen 600 documents (300 training documents and 300 testing documents) as indicated in Table 1.

TMC2007: This is the dataset used for the SIAM 2007 competition organized by the text mining workshop held in conjunction with the 7th SIAM International Conference on Data Mining [25]. This competition sponsored by NASA Ames Research Center, focused on developing text mining algorithms for document classification. It contains 28596 aviation safety reports in free text form, annotated with one or more out of 22 problem types that appear during certain flights [26]. However, in order to save computational resources, we have randomly chosen 300 training documents and 300 testing documents for our experiments. The dataset comes from human generated reports on incidents that occurred during the flights which means there is one document per incident. Text representation follows the boolean bag-of-words model. The goal was to label the documents with respect to the types of problems that were described. This is a subset of the Aviation Safety Reporting System (ASRS) dataset, which is publicly available. Some other statistics of the dataset are given in Table 1.

Table 1. The main characteristics of the selected subset of the datasets.

problem	samples	nominal	numeric	label	cardinality	density	distinct
rcv1v2	600	0	47235	103	2.642	0.026	946
tmc2007	600	49060	0	22	2.158	0.098	1341

In the applications using text categorization as the core task, the computational efficiency is crucial because of very large number of features, classes and samples. Therefore, the need for designing a simple and fast classification system is important. There are many research studies using different kinds of classifiers such as k-nearest neighbors (kNN), support vector machines (SVM), artificial neural networks (ANN), bayesian methods and rocchio classifiers [3]. However, in practice most of them are not applicable as in real-world applications, e.g. search engines and recommender systems, a just-in-time response has great importance. Among them, the naive bayes and centroid classification algorithms are extremely simple and straightforward illustrating competitive performance on text categorization problems. Moreover, they do not need to memorize a huge amount of training data as some other classifiers do (e.g. kNN) and adjust so many parameters (e.g. ANN).

For the experiments presented in the current paper, we used *centroid-based classifiers* as the ECOC dichotomizers. This means that the prototype vector or centroid vector (μ_i^+) is computed for super-class $\mathcal{T}_i^+$ as:

$$\mu_i^+ = \frac{1}{\mid \mathcal{T}_i^+ \mid} \sum_{d \in \mathcal{T}_i^+} d \tag{3}$$

where $\mid \mathcal{T}_i^+ \mid$ denotes the cardinality of set $\mathcal{T}_i^+$, i.e. the number of documents that belong to positive set in the i-th individual and d is a training document.

In the testing step, we calculate the similarity of a document d to each centroid by the *cosine* measure,

$$S(d, \mu_i^+) = \frac{d \cdot \mu_i^+}{\mid\mid d \mid\mid \ \mid\mid \mu_i^+ \mid\mid} \tag{4}$$

This similarity can be regarded as the *posterior probability* of the dichotomizer and used for i-th bit of the predicted codeword $\bar{y}_d$.

Consequently, the evaluation of methods to handle multi-label data requires different measures than those used for traditional single-label classification. Various measures are traditionally being used for evaluation of multi-label classification (particularly for document and text applications) such as *classification accuracy*, *precision*, *recall* and *F1*. These are defined below.

$$classification\ accuracy = \frac{1}{n} \sum_{d=1}^{n} \mathcal{I}(\omega_d = \bar{\omega_d}) \tag{5}$$

where $I(true) = 1$ and $I(false) = 0$ and n is the number of documents in a dataset. This is a very strict evaluation measure as it requires the predicted set to be an exact match for the true set in the label set no matter if a classifier makes a mis-classification at only one category or the entire set.

$$precision = \frac{1}{N_c} \sum_{c_i=1}^{N_c} \frac{TP_{c_i}}{TP_{c_i} + FP_{c_i}} \ \ and \ \ recall = \frac{1}{N_c} \sum_{c_i=1}^{N_c} \frac{TP_{c_i}}{TP_{c_i} + FN_{c_i}} \tag{6}$$

where TP, FP and FN stand for the true positive, false positive and false negative for each category, respectively. The F1-score which considers both the *precision* and *recall* of the test set is formulated as:

$$F1 = \frac{2precision.recall}{precision + recall} \tag{7}$$

where an $F1$ score reaches its best value at 1 and worst score at 0.

We have compared the results of the proposed method with some of commonly used TC algorithms. The standard multi-label TC methods used as baseline methods are the big-bang (global method) and Local Classifier per Category (LCC). For all these methods, centroid-based classifiers with the same parameters have been implemented. As shown in Table 2, the proposed ML-ECOC using Dense random and 2vsA codes outperforms the standard TC approaches on the selected datasets by obtaining the maximum F1 scores. One can note that the results for 2vsA code for rcv1v2 data is missing. This is because of large number of classes of RCv1v2 data which make building ECOC classifier unfeasible.

To give more detailed information, Figure 3 shows precision-recall curves corresponding to ML-ECOC and LCC approaches. Because of the superior performance on ML-TC datasets, the LCC approach is used for assessing the comparative performance of ML-ECOC. As clearly shown, the proposed ML-ECOC is able to obtain slightly better results on RCv1-v2 while always winning on TMC2007 data.

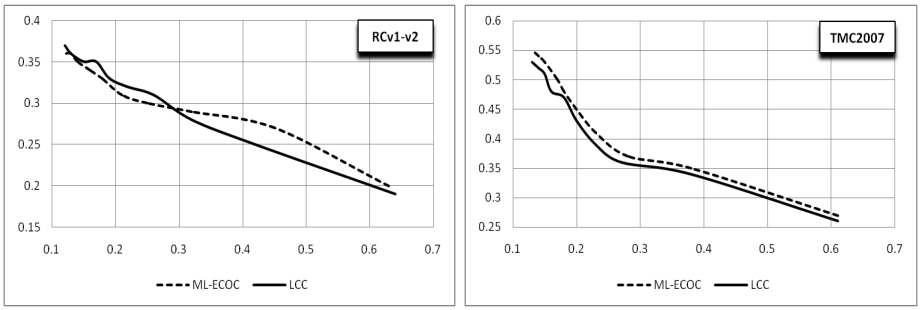


Fig. 3. Precision-Recall curves for the RCV data (left) and TMC2007 (right). X-Y axis represent the precision and recall, respectively.

5 Conclusions

An extension of the ECOC algorithm called ML-ECOC is proposed to tackle multi-label TC problems. To avoid the inconsistency in coding step, the proposed ML-ECOC method decomposes a multi-label problem into some complementary one-class sub-problems unlike the standard ECOC which builds dichotomies.

Table 2. F1 score of the proposed method (PM) using different coding strategies compared to the existing standard text categorization methods on the selected subset of rcv1v2 and tmc2007 datasets (F1 values are reported in percentage).

Problem	big-bang	LCC	ML-ECOC (drand)	ML-ECOC (2vsA)
rcv1v2	37.5	30.1	**32.9**	32.7
tmc2007	31.3	35.7	34.9	**36.5**

Multi-label relationship is taken into account in the testing phase by using a novel decoding strategy adopted for ECOC algorithm. Experimental results on Reuters datasets confirm the potential of the proposed ML-ECOC on multi-label classification with large number of categories.

Recently, some studies [23, 24] try to increase ECOC reliability by proposing a reject mechanism. One interesting future research line refers to multi-label text categorization with a reject option.

Acknowledgments. Camelia Chira acknowledges the support of Grant PN II TE 320, Emergence, auto-organization and evolution: New computational models in the study of complex systems, funded by CNCS Romania.

References

1. Maron, M. 1961. Automatic indexing: an experimental inquiry. J. Assoc. Comput. Mach. 8, 3, 404-417.
2. Lin S. and Costello. D. J. Error Control Coding, Second Edition. Prentice-Hall, Inc. (2004)
3. F. Sebastiani, Machine learning in automated text categorization, ACM Computing Surveys, Volume 34 Issue 1, 2002.
4. SABLE, C. L., Hatzivassiloglou, V. 2000. Textbased approaches for non-topical image categorization. Internat. J. Dig. Libr. 3, 3, 261-275.
5. Lewis, D. D., Yang, Y., Rose, T. G., and Li, F. (2004). RCV1: A new benchmark collection for text categorization research. Journal of Machine Learning Research, 5, 361-397
6. Schapire, R. E., Singer, Y. 2000. BoosTexter:a boosting-based system for text categorization. Mach. Learn. 39, 2/3, 135-168.
7. R. Ghani, Using Error-Correcting Codes for Text Classification, 17th International Conference on Machine Learning, 2000.
8. A. Berger, Error-Correcting Output Coding for Text Classification, In Proceedings of IJCAI-99 Workshop on Machine Learning for Information Filtering, 1999.
9. Pujol, O. Radeva, P. Vitria, J. Discriminant ECOC: A heuristic method for application dependent design of error correcting output codes, IEEE Transactions on PAMI 28 (6), 1001-1007 (2006)
10. J. Zhou, H. Peng, C. Y. Suen, Data-driven decomposition for multi-class classification, Pattern Recognition, 41 67-76 (2008)
11. Hatami, N. Thinned-ECOC ensemble based on sequential code shrinking. Expert Systems with Applications. 39 (2012) 936-947.

12. Hastie, T. Tibshirani, R., Classification by pairwise grouping, The Annals of Stat. 26 (5), 451-471 (1998)
13. Jordan, M. I. and R. A. Jacobs (1994). Hierarchical mixtures of experts and the EM algorithm. Neural Computation 6(2), 181-214.
14. T. G. Dietterich and G. Bakiri. Solving multiclass learning problems via error-correcting output codes. Journal of Arti?cial Intelligence Research, 2:263-286, 1995.
15. Breiman, Leo (2001). "Random Forests". Machine Learning 45 (1): 5-32.
16. Leo Breiman (1996). Bagging predictors. Machine Learning 24 (2): 123-140.
17. Yoav Freund and Robert E. Schapire A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences, 55(1):119-139, 1997.
18. Ho, Tin (1998). "The Random Subspace Method for Constructing Decision Forests". IEEE Transactions on Pattern Analysis and Machine Intelligence 20 (8): 832-844.
19. Baoli Li, Carl Vogel: Improving Multiclass Text Classification with Error-Correcting Output Coding and Sub-class Partitions. Canadian Conference on AI 2010: 4-15
20. Songbo Tan, Gaowei Wu, Xueqi Cheng: Enhancing the Performance of Centroid Classifier by ECOC and Model Refinement. ECML/PKDD (2) 2009: 458-472
21. G. Tsoumakas, I. Katakis, Multi-Label Classification: An Overview, International Journal of Data Warehousing and Mining, 3(3):1-13, 2007
22. Wei Bi, James T. Kwok: MultiLabel Classification on Tree- and DAG-Structured Hierarchies. ICML 2011: 17-24
23. G. Armano, C. Chira, and N. Hatami, Ensemble of Binary Learners for Reliable Text Categorization with a Reject Option, HAIS 2012, in press
24. Paolo Simeone, Claudio Marrocco, Francesco Tortorella: Design of reject rules for ECOC classification systems. Pattern Recognition 45(2): 863-875 (2012)
25. http://www.cs.utk.edu/tmw07/
26. A. Srivastava and B. Zane-Ulman. Discovering recurring anomalies in text reports regarding complex space systems. In IEEE Aerospace Conference, 2005.

How well do we know Bernoulli?

Giorgio Maria Di Nunzio[1] and Alessandro Sordoni[2]

[1] Dept. of Information Engineering – University of Padua
dinunzio@dei.unipd.it
[2] Dept. of Computer Science and Operations Research – University of Montreal
sordonia@iro.umontreal.ca

Abstract. Naïve Bayes probabilistic models are widely used in text categorization because of their efficient model training and good empirical results. Bayesian classifiers face a common issue called data sparsity problem which makes an adequate estimation of probabilities a difficult task. Therefore, smoothing techniques are needed in order to adjust the maximum likelihood estimators. In this preliminary paper we make use of a visualization technique to further investigate the expressiveness of the well known Bernoulli Naïve Bayes classifier. Various smoothing methods are tested by means of a visual analysis which makes the estimation of optimal parameters straightforward. Experimental results demonstrated that: (1) visual analysis is a valuable tool for understanding the behaviour of smoothing methods and their limits (2) the Bernoulli multivariate model performance can increase significantly with a suitable setting of smoothing parameters.

1 Introduction

A large number of studies have shown that Support Vector Machines (SVM) can outperform other approaches in many categorization applications [1], but Naïve Bayes (NB) is still widely used in practice mostly likely due to its tradeoff between very efficient model training and good empirical results. NB classifiers are sensitive to the data sparsity problem which is particularly evident when the size of training data is small. Due to data sparseness, the maximum likelihood estimation of the probability of unseen features (terms in the case of text classification) tend to be zero. To prevent this undesirable behaviour, smoothing techniques are a possible solution. Smoothing a probability actually means assigning a non-zero probability to the features that describe the object we want to classify. Several smoothing methods have been proposed [2]: additive, or *Laplacian* smoothing, Jelinek-Mercer, Dirichlet, absolute discount and two-stage smoothing. Some of these approaches operate an interpolation with a background collection model, some others simply add extra counts to the observed frequency of each feature.

In this preliminary work, we are interested in studying smoothing methods for the multi-variate Bernoulli classifier. Most research so far has shown that the multinomial Naïve Bayes generally outperforms the Bernoulli classifier both in text categorization [3] and information retrieval [4]. From a probabilistic point

of view, the latter model makes a weaker independence assumption about word occurrences at the price of not being able to model multiple word occurrences. Even if there has been some empirical evidence that multinomial outperforms multi-variate Bernoulli, the need for a more systematic comparison between these model is needed [5]. Therefore, we put forward the following research question: how far can we improve the performance of the Bernoulli classifier by setting optimal Beta prior smoothing parameters? The objective of our experimental evaluation (inspired by the work of [2]) is to compare three well-established smoothing methods against a manual optimization of the Beta parameters by means of the two-dimensional visual approach [6].

2 Bayesian and Jelinek-Mercer Smoothing

Given a set C of categories, the bayesian approach to categorization consists by estimating $\mathrm{P}(d|c_i)$ and calculating the posterior $\mathrm{P}(c_i|d)$ via Bayes rule[3]. The multi-variate Bernoulli model represents a document as a binary vector over the space of terms in which each dimension indicates whether the term occurs in the document. The occurrence of each term is governed by a Bernoulli distribution. Learning the parameters of this model corresponds to estimating class-conditional Bernoulli parameters $\theta_{t_k|c_i} \equiv \mathrm{P}(t_k|c_i;\theta)$, where t_k is a term of the vocabulary. The maximum likelihood (ML) estimators of this parameters are of the form:

$$\hat{\theta}^{ML}_{t_k|c_i} = \frac{\tau_{k,i}}{m_i} \tag{1}$$

where $\tau_{k,i}$ is the number of documents belonging to c_i in which term t_k appears and m_i is the total number of documents in c_i. The ML is zero for terms that never occur in documents in c_i. To prevent this undesirable behavior, the choice of a suitable prior to smooth probabilities is a possible solution. The conjugate prior of the Bernoulli distribution is the beta-distribution $beta(\theta;\alpha,\beta)$, where α and β are hyper-parameters. Assuming this prior, the smoothed estimate of the probability of a term t_k given a category c_i is given by the posterior mean [7]:

$$\hat{\theta}^{B}_{t_k|c_i} = \frac{\tau_{k,i} + \alpha}{m_i + \alpha + \beta} \ , \tag{2}$$

Setting $\alpha = 1, \beta = 1$ is called Laplace smoothing. Using the Jelinek-Mercer (JM) method, this parameter is computed by interpolating the maximum likelihood estimate with a collection language model $\theta_{t_k|C} \equiv \mathrm{P}(t_k|C;\theta)$:

$$\hat{\theta}^{ML}_{t_k|C} = \frac{\tau_k}{m} \ , \tag{3}$$

where τ_k is the number of documents in which term t_k appears and m the number of documents in the collection. Using λ as the interpolation parameter, the Jelinek-Mercer can be written as:

$$\hat{\theta}^{JM}_{t_k|c_i} = (1-\lambda)\hat{\theta}^{ML}_{t_k|c_i} + \lambda\hat{\theta}^{ML}_{t_k|C} \ , \tag{4}$$

[3] $\mathrm{P}(c_i|d) = \mathrm{P}(d|c_i)\mathrm{P}(c_i)/\mathrm{P}(d)$, where $c_i \in C$ and d is a document.

with $0 \leq \lambda \leq 1$. For $\lambda = 0$, we obtain the maximum likelihood estimator, while for $\lambda = 1$ we completely rely on the collection language model. Indeed, opposite to Beta smoothing, the Jelinek-Mercer smooths each parameter $\hat{\theta}^{ML}_{t_k|c_i}$ by a different amount depending on the probability of the term with respect to the entire collection. Nevertheless, looking closer at Eq. (2), we can write:

$$\hat{\theta}^{B}_{t_k|c_i} = \frac{m_i}{m_i + \alpha + \beta} \frac{\tau_{k,i}}{m_i} + \frac{\alpha + \beta}{m_i + \alpha + \beta} \frac{\alpha}{\alpha + \beta} \ , \tag{5}$$

which means that the probability of a term is obtained by interpolating the maximum likelihood estimator with the prior mean $\alpha/(\alpha + \beta)$. Setting $\alpha = \beta\, \tau_k/m - \tau_k$, such that $\alpha/(\alpha + \beta) = (\tau_k/m)$, we recover the JM except that the interpolation slope is flatter: we must allow β to vary through a bigger interval in order to recover JM estimations[4]. In our experiments, and for the purpose of visual analysis, we limit ourselves to use the same α and β for each smoothed estimate of term t_k. As we will see in the next sections, this will represent a lack of expressiveness of the Beta prior smoothing and opens a path for the continuation of this work.

3 Visualization of Priors' Effects

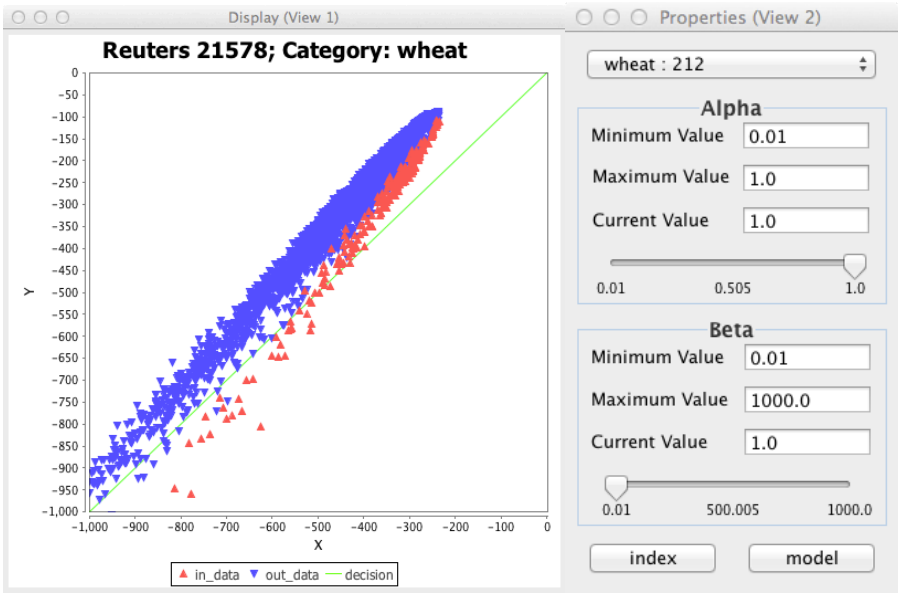
In this work, we make use of a visual analysis tool, namely the two-dimensional visualization of probabilistic models [6], for understanding the behaviour of smoothing methods and their limits. In the two-dimensional visualization, two coordinates are calculated for each document d and for each category c_i. These two coordinates correspond to the two posterior probabilities $\mathrm{P}(c_i|d;\hat{\theta})$ and $\mathrm{P}(\bar{c}_i|d;\hat{\theta})$ governed by the estimated parameter $\hat{\theta}$. We compare these two probabilities to decide whether the document belongs to c_i or not. By applying Bayes rule and taking the logs in order to avoid arithmetical anomalies (products of very small numbers tend to zero very quickly) we obtain:

$$\log\Big(\mathrm{P}(d|c_i;\hat{\theta}_{c_i})\Big) + \log\Big(\mathrm{P}(c_i;\hat{\theta})\Big) > \log\Big(\mathrm{P}(d|\bar{c}_i;\hat{\theta}_{\bar{c}_i})\Big) + \log\Big(\mathrm{P}(\bar{c}_i;\hat{\theta})\Big) \tag{6}$$

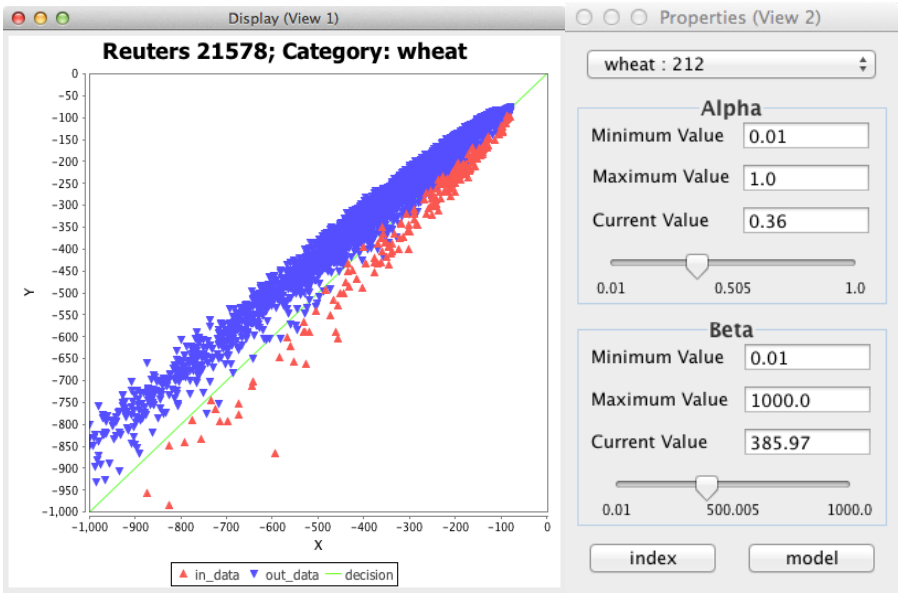
Given a category c_i, each coordinate of a document is the sum of two addends: a variable component which depends on the terms that appear in the document, and a constant component related to probability of the category itself. The probability $\mathrm{P}(d|c_i;\hat{\theta}_{c_i})$ is in turn estimated by combining the estimates $\hat{\theta}_{t_k|c_i}$ for each term in the document. We can therefore determine and change the position of the document in the two-dimensional space by adjusting the hyper-parameters α and β.

An example of this visualization is shown in Figure 1. The decision boundary is represented by the green line: below the line, the document is assigned to the category c_i, above the line, the document is assigned to $\bar{c}_i$. The influence of a change in the values α and β is visualized with an animation of the documents in the space.

[4] A similar derivation is done in [8] with a Dirichlet prior.



(a) Display window. (b) Properties window



(c) Display window. (d) Properties window

Fig. 1: Two-dimensional tool display for category "wheat" of REUTERS-21578 collection. Figure 1a and 1b show the distribution of documents for $\alpha = 1$, $\beta = 1$, Laplace smoothing. Figure 1c and 1d show the distribution of documents for a different setting of the parameters. The red triangles $\triangle$ are the documents of the category to classify, the blue diamonds $\bigtriangledown$ are all the other documents of the collection. The decision frontier is drawn in green.

		Reuters-21578			20-newsgroups			Ohsumed		
Average		Recall	Precision	F1	Recall	Precision	F1	Recall	Precision	F1
Macro	(la)	0.341	0.418	0.350	0.047	0.269	0.076	0.105	0.531	0.138
	(je)	0.745	0.669	0.701	**0.749**	0.731	**0.727**	**0.636**	0.549	**0.577**
	(ey)	0.751	0.542	0.622	0.707	0.612	0.610	0.475	**0.557**	0.494
	(vi)	**0.798**	**0.717***	**0.749***	0.708	**0.755**	0.723	0.480	0.550	0.500
micro	(la)	0.672	0.661	0.666	0.047	0.292	0.047	0.235	**0.699**	0.351
	(je)	0.857	0.785	0.820	**0.752**	0.717	**0.736**	**0.659**	0.553	**0.601**
	(ey)	0.869	0.644	0.740	0.713	0.517	0.600	0.578	0.581	0.579
	(vi)	**0.879**	**0.841**	**0.860**	0.715	**0.755**	0.734	0.579	0.581	0.580

Table 1: Comparison of micro and macro average Precision, Recall and F1 measure for three of the four smoothing methods tested on the considered collections. The best performance is highlighted in bold. The star denotes a statistical significant improvement of the measure according to the Wilcoxon test applied to the vectors of scores on each category with the alpha value of 5%

4 Experiments

We tested Jelinek-Mercer (je) against three different parametrization of the beta prior: (la) a uniform (Laplace smoothing) beta prior, $beta(\theta; 1, 1)$; (ey) a beta distribution was set as found by Eyheramendi et al. [9], $beta(\theta; 0.1, 0.3)$; (vi) a beta distribution with optimal parameters α^* and β^*, $beta(\theta; \alpha^*, \beta^*)$. We found λ^* and α^*, β^* for each category by optimizing the F1-score (F1) on the training set: the Jelinek-Mercer λ^* was selected by iterative searching over the interval $[0, 1]$; for α^*, β^*, we exploited the document visualization technique. As overall quality measures, we used standard ATC micro- and macro-averged Recall, Precision, and F1 measures [1].

We selected three of the most widely used collections in literature. We tested Reuters-21578 using the 10 most frequent categories following the "ModAptè" split (9,603 training and 3,299 test documents); 20 Newsgroups, 20 categories with 18,846 stories, divided in 60%-40% training/test; Ohsumed, 6,286 training and 7,643 for test documents classified into 23 Medical Subject Headings (MeSH). These subsets of the collections were chosen accordingly to most of the literature in Automated Text Categorization (ATC) [3, 1, 10]. Default English stopwords were removed and all letters have been converted to lowercase. The two-dimensional interface was implemented in Java using Java Swing technologies.

The baseline obtained by (la) performed statistically worse than any other approach upon the considered datasets: Church and Gale presented strong arguments against the effectiveness of *add-one* smoothing for language data in [11]. As we started the visual search from the parameters set by (ey), (vi) cannot be worse than (ey). Nevertheless, since the parameters found with (vi) were optimized by monitoring the F1 measure, it may happen that with a higher F1, either the value of Recall or Precision are less than (ey). The averaged results on the three datasets are reported in Table 1.

Visual parameter optimization significantly improves categorization performances over the three methods in REUTERS. Fig. 1 illustrates how visual optimization operates for the category "wheat". Applying the same amount of smoothing to each term reveals to be effective in this collection: almost all categories are well represented and using the collection language model as an evidence source for smoothing is not of much interest. Nevertheless, by taking a closer look to performances on each category (not reported in this paper), we found indeed that JM performs best on difficult categories (SHIP, WHEAT). This tendency is clearly emerging on the other two collections. On 20 NEWSGROUPS, visual optimization greatly increases Precision performances over static (EY) parameters. Despite this fact, Beta prior smoothing with optimal parameters reaches the same expressiveness as Jelinek-Mercer (JM) smoothing. On the OHSUMED collection, visual optimization confirms that Beta prior smoothing is lacking expressiveness for this dataset. Computing the mean and the variance of the optimal λ^* parameter found for each category we obtained $\mu_{\lambda^*} = 0.82$, $\sigma^2_{\lambda^*} = 0.02$ thus confirming that taking evidence at a collection level is relevant when dealing with noisy documents and semantically overlapping categories.

5 Conclusions

In this preliminary work, we have studied the effects of smoothing methods for the NB classifier by means of visualization analysis. In the initial phase of this research, we have focused our analysis on the simplest NB model: the multi-variate Bernoulli model. We put forward the following research question: how far can we improve the performance of the Bernoulli classifier by setting optimal Beta prior smoothing parameters? The objective of our experimental evaluation was to compare three well-established smoothing methods against a manual optimization of the Beta parameters (which govern the smoothing of the probabilities) by means of the two-dimensional visual approach.

Experiments have shown that it is possible to find hyper-parameters of the Beta prior that improve the classification significantly. However, in this first set of experiments we limited ourselves to the use the same α and β for each term. A natural continuation of this research will be to find an automatic way to estimate different α and β parameters for each term and to understand if this actually improves performance measures. This problem will consist in characterizing the first and second order moment of each Beta prior distribution based on some relevant empirical evidence of term occurrence in the collection.

This initial set of experiments will lead to the second phase of the study: the analysis of the smoothing methods for the multinomial NB model. Most research so far has shown that the multinomial Naïve Bayes generally outperforms the Bernoulli classifier both in text categorization and information retrieval. From a probabilistic point of view, the Bernoulli model makes a weaker independence assumption on word occurences. This is why we believe that a more systematic comparison between these model is still needed [5]. Another thread of research will be to apply the visualization analysis to more complex NB models, such

as the Chain Augmented NB models (also known as CAN models) which allow a straightforward the application of sophisticated smoothing techniques from statistical language modeling [10].

Acknowledgments. This work has been partially supported by the QONTEXT project under grant agreement N. 247590 (FP7/2007-2013).

References

1. Sebastiani, F.: Machine learning in automated text categorization. ACM Comput. Surv. **34** (2002) 1–47
2. Zhai, C., Lafferty, J.D.: A study of smoothing methods for language models applied to information retrieval. ACM Trans. Inf. Syst. **22** (2004) 179–214
3. McCallum, A., Nigam, K.: A comparison of event models for naive bayes text classification. In: AAAI-98 workshop on learning for text categorization. Volume 752. (1998) 41–48
4. Metzler, D., Lavrenko, V., Croft, W.B.: Formal multiple-bernoulli models for language modeling. In: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval. (2004) 540–541
5. Zhai, C.: Statistical language models for information retrieval. Synthesis Lectures on Human Language Technologies **1** (2008) 1–141
6. Di Nunzio, G.: Using scatterplots to understand and improve probabilistic models for text categorization and retrieval. Int. J. Approx. Reasoning **50** (2009) 945–956
7. Gelman, A., Carlin, J.B., Stern, H.S., Rubin, D.B.: Bayesian Data Analysis, Second Edition (Chapman & Hall/CRC Texts in Statistical Science). 2 edn. Chapman and Hall/CRC (2003)
8. Smucker, M.D., Allan, J.: An investigation of dirichlet prior smoothing's performance advantage. Technical Report Technical Report IR-391, The University of Massachusetts, The Center for Intelligent Information Retrieval (2005)
9. Eyheramendy, S., Lewis, D.D., Madigan, D.: On the Naive Bayes Model for Text Categorization. In Bishop, C., Frey, B., eds.: Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics. (2003)
10. Peng, F., Schuurmans, D., Wang, S.: Augmenting naive bayes classifiers with statistical language models. Inf. Retr. **7** (2004) 317–345
11. Gale, W.A., Church, K.W.: What's wrong with adding one? In: Corpus-Based Research into Language. Rodolpi. (1994)

Investigating the Use of Extractive Summarisation in Sentiment Classification

Marco Bonzanini, Miguel Martinez-Alvarez, and Thomas Roelleke

Queen Mary, University of London
Mile End Road, E1 4NS London, UK
{marcob,miguel,thor}@eecs.qmul.ac.uk

Abstract. In online reviews, authors often use a short passage to describe the overall feeling about a product or a service. A review as a whole can mention many details not in line with the overall feeling, so capturing this key passage is important to understand the overall sentiment of the review. This paper investigates the use of extractive summarisation in the context of sentiment classification. The aim is to find the summary sentence, or the short passage, which gives the overall sentiment of the review, filtering out potential noisy information. Experiments are carried out on a movie review data-set. The main finding is that subjectivity detection plays a central role in building summaries for sentiment classification. Subjective extracts carry the same polarity of the full text reviews, while statistical and positional approaches are not able to capture this aspect.

1 Introduction

The popularity of on-line resources, which allow users to review products or services, is motivating new interest in the area of Sentiment Analysis [10]. One of the main tasks in this field is the classification of opinionated documents according to the overall sentiment, i.e. whether positive or negative. A common behaviour among reviewers is to summarise the overall sentiment of the review in a single sentence, or in a short passage. On the other hand, the rest of the review can express a feeling which is different from the overall judgement. This can be explained by the presence of several aspects or features that the reviewers want to comment on. As an example, we can consider the following review, taken from RottenTomatoes[1], a popular movie review site. The words or phrases carrying opinions are marked in italic. Several sentences express disappointment about different aspects of the movie, and simply counting the negative sentences would lead to classify the review as negative. The overall recommendation, described in the last sentence, is instead positive. It is also worth noting that some expressions, like "too easily", do not carry a negative sentiment per se, but must be put into context to be understood. In a similar way, terms normally related to negative feelings, like "trauma", are not used to denote a negative opinion:

[1] http://www.rottentomatoes.com

> I was particularly *disappointed* that the film didn't deal more with the trauma of learning one's life is a tv show [...] I almost felt that he got over it *too easily* for the sake of the film's pacing [...] Perhaps it's not fair to criticize a movie for what it isn't, but it seems like there were *some missed opportunities* here. But on its own terms, the movie is *well made*.

Moreover, often a review contains sentences which do not provide any information about opinions, i.e. they are not subjective. This is the case of movie reviews, where a short picture of the plot can be given to open the review, without commenting on it. Previous work has shown how the capability of identifying subjective sentences can improve the sentiment classification [9].

This paper investigates how the use of summarisation techniques can be applied in the context of sentiment classification of on-line reviews. The focus is on the movie review domain, which is considered to be particularly challenging, as people write not only about the movie itself, but also about movie elements such as special effects or music, and about movie-related people [15]. More specifically, the aim is to capture the summary passage, i.e. the short passage, or even the single sentence, which gives the overall sentiment of the review. From the user's perspective, the advantage of having a summarised review consists in a reduced effort to understand the message of the document, given that the key information is preserved. Traditional sentence extraction techniques can be applied for this task, although a more opinion-oriented approach is needed, since the goal is not to better describe the topic of the review in a single sentence, but to capture its overall polarity. In order to verify whether the summarisation task preserves the information about the sentiment of reviews, text classification is performed on the original documents and on the produced summaries.

The contributions of this work are two-fold: firstly, we show how the summaries based on subjectivity well represent the polarity of the full-text review; secondly, we investigate different techniques for identifying the key passage of a review with respect to polarity. Experiments on a movie review data-set show the importance of subjectivity detection for polarity classification.

The rest of the paper is organised as follows. Section 2 presents the related work on sentiment summarisation, and classification through summarisation. In Section 3 the overall approach for sentiment classification and summarisation is proposed. Section 4 reports the experimental study, and Section 5 concludes the paper outlining the directions for future work.

2 Related Work

Previous work in summarisation of opinionated documents has been focusing on different domains of user-generated content. Dealing with short web comments, an approach for extracting the top sentiment keywords and for showing them in a tag cloud, has been proposed in [11]. This approach is based on the use of Pointwise Mutual Information (PMI) as described in [14]. Experiments in the context of digital product reviews have

been reported in [4]. This technique uses a set of seed adjectives of known polarity, which is expanded with the use of WordNet (i.e. synonyms share the same polarity, while antonyms have the opposite polarity). The generation of summaries consists then in aggregating opinionated sentences related to the same feature. A multi-knowledge approach has been shown in [15], with experiments on the movie review domain. This approach aims at identifying movie features, like the soundtrack or the photography, as well as movie-related people, like actors, director, etc. Since single opinions can be expressed on a specific feature of a movie, their approach can be used to build personalised feature-oriented summaries. A similar work has been proposed in [2] in the context of local service reviews.

The use of summarisation to improve classification has been explored in [13]. Different summarisation techniques can be applied to generate summaries of web-page, resulting in an improvement of their classification. This approach differs from the one proposed in this paper, as they face the problem of topic classification rather than sentiment classification. The work presented in [3] implies the use of sentence extraction techniques, although it is not focused on summarisation per se. The use of sentence-level evidences, in particular the location of the sentence within the document, is used to improve opinion retrieval. In this approach, relevance and polarity are combined to retrieve blog posts.

The idea of a single sentence extraction, to determine the polarity of the whole document, has been suggested in [1], although results on the polarity classification task have not been reported. Another summarisation approach, based on subjectivity detection, is shown in [9]. The main idea is to filter out the objective sentences, i.e. the ones not carrying sentiment information, and to base the polarity classification entirely on the subjective sentences. Proximity information is also taken into account, as subjective sentences tend to be close to each other. This method has been shown to significantly improve the classification, compared to the results of a Naive Bayes classifier on the whole document, and to be not significantly worse than a Support Vector Machine classifier.

3 Methodology

This section describes the main components of the proposed approach, namely a sentiment classifier, an extractive summariser and a subjectivity classifier. Figure 1 describes the pipeline for the movie review classification. The reviews can be classified directly (full text) or can be summarised in three different ways. Firstly, through the summarisation component, sentence extraction based on statistical or positional approaches is performed. Secondly, through the subjectivity detection component, objective sentences are filtered out, keeping all and only the subjective ones to form the summary. Thirdly, through a pipeline of both components, subjective extracts are further summarised.

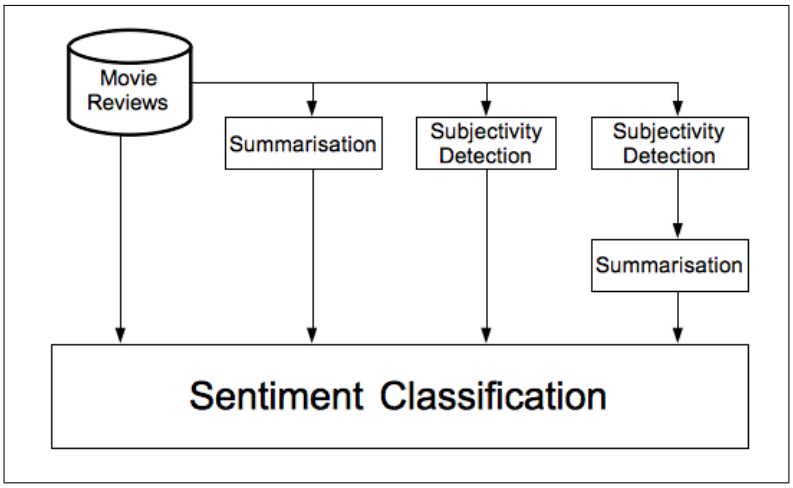


Fig. 1. Pipeline of the review summarisation and classification

3.1 Sentiment Classification

Sentiment classification is a text classification task, where a label indicates the polarity of the document rather than its topic. The task can be approached from different points of view. For example, identifying the overall sentiment of a document is different from mining the polarity of individual aspects like soundtrack, plot, etc. In this paper, only the polarity of the document as a whole is considered, i.e. whether the overall recommendation of a review is positive or negative.

Traditional machine learning approaches can be applied for this classification task. Specifically, Naive Bayes (NB) and Support Vector Machine (SVM) classifiers are considered, using unigram-presence as features. The feature selection for NB is based on document frequency, being a commonly used selection strategy.

3.2 Extractive Summarisation

In order to produce different kinds of extractive summaries, different sentence selection techniques are applied. Notice that using unigrams as features for the classification, rebuilding the original order of the sentences is necessary only when a further summarisation step, which considers sentence position or proximity, is performed.

The considered techniques are the following:

- Luhn's traditional approach, as representative of statistical approaches;
- positional approaches, based on the intuition that the location of the sentence within the document reflects its significance;
- subjectivity detection, used to filter out sentences which do not express opinions;
- combinations of subjectivity detection with the other approaches.

Luhn's approach Firstly, the traditional Luhn's approach [6] is used to score the sentences according to their significance. The top N sentences are selected to create the summary. The results for this approach are labelled as *Luhn-N*, where N is the number of sentence used to create the summary. The significance score of a sentence is based on clustering of sentence tokens using a distance threshold (5 is the distance used in this paper). For each cluster, the score is computed taking the ratio between the square of the number of significant words in the cluster, over the total number of words in the cluster. The significant words are chosen according to their frequency, i.e. the terms with higher tf, excluding stop words, are considered significant. The significance score for a sentence will be the maximum score for any of its clusters.

Position-based approaches A second family of summarisers is built on top of an empirical observation: often reviewers tend to summarise their overall feeling in a sentence or in a short paragraph, placed either at the beginning or at the end of the review. In this case, a summary can be created simply selecting the N opening sentences, or the N closing sentences. Results for these approaches are labelled as *First-N* and *Last-N*, respectively.

Subjectivity detection The previous approaches do not take into account the subjective nature of the documents under analysis. To overcome this issue, the aforementioned classification techniques can be used to identify and filter subjective sentences. A specific data-set, described in Section 4, is used to train the classifiers. Filtering out the objective sentences and aggregating only the subjective ones can already be seen as a summarisation approach. The average compression rate of the data under analysis is around 60%. Results for this approach are labelled as *Subjective-Full*.

Summarising subjective extracts In order to further increase the compression rate, and to take into account subjectivity, one of the first two approaches can be applied to the subjective extracts. In the results, this family of approaches is labelled as follows: *Subjective-Luhn-N* for the summaries produced using Luhn's approach on the subjective sentences, *Subjective-First-N* and *Subjective-Last-N* for the summaries based on the subjective sentence positions. Again, N represents the number of selected sentences.

4 Experimental Study

The evaluation of summarisation systems is a research issue in itself, and different intrinsic evaluation approaches have been proposed over the years [7]. Since the purpose of this work is observing how the use of summarisation techniques can help the sentiment classification task, we do not evaluate the summaries with traditional methods like ROUGE [5] or Pyramid [8], nor we look for linguistic quality. The evaluation is performed with respect to the polarity classification, i.e. a good summary is ideally able to carry the same polarity of the full document. Full text reviews and summaries are classified according to their overall polarity.

4.1 Experimental Setup

For the subjectivity detection, a data-set of subjective and objective sentences is used to train the classifiers [9]. This data-set contains 5000 subjective sentences, taken from RottenTomatoes snippets, and 5000 objective sentences, which are taken from IMDb plots. The main idea behind the creation of the subjectivity data-set consists in assuming that the review snippets from RottenTomatoes contain only opinionated sentences, while the movie plots taken from IMDb contain non-opinionated, and hence objective, sentences. Firstly, the classifiers are tested on the subjectivity data-set, using a five-folding cross-validation approach. The micro-averaged F_1 results are not significantly different (88.85 for NB vs. 88.68 for SVM). The classifiers can be considered reliable enough for the subjectivity detection task which leads to the generation of subjective extracts.

The sentiment classification has been evaluated on the movie review data-set firstly used in [9], containing reviews taken from IMDb[2] and annotated as positive or negative. The data-set contains 2000 documents, evenly distributed between the two classes.

4.2 Results and Discussion

Table 1 reports the results of the micro-averaged F_1 scores on the review data-set. This evaluation measure is chosen as it is one of the most commonly used in text classification [12]. The macro-averaged results are not reported as they are very similar to the micro-averaged ones, given the data-set is well balanced, i.e. the two classes contain the same number of document.

Table 1. The micro-averaged F_1 results of sentiment classification

	NB	SVM		NB	SVM
Full Review	83.31	87.10	Subjective-Full	84.61	86.82
Luhn-1	70.12	70.28	Subjective-Luhn-1	71.02	70.50
Luhn-3	75.47	74.96	Subjective-Luhn-3	74.92	74.91
First-1	68.94	68.82	Subjective-First-1	69.33	68.90
Last-1	70.61	70.49	Subjective-Last-1	70.90	71.15
First-3	70.81	70.43	Subjective-First-3	71.12	71.07
Last-3	75.58	76.57	Subjective-Last-3	75.49	76.26

The first observation is that statistics and positional summarisation approaches do not provide any improvement to the sentiment classification results. On the contrary, the performances are substantially worse for both NB and SVM. The explanation behind this behaviour is that these approaches are not explicitly opinion-oriented, so they are not able to capture the sentiment behind a review.

[2] http://www.imdb.com

The quality of sentiment classification for subjective extracts is instead in line with the full review classification. More precisely, the classification of subjective extracts through NB achieves a 1.5% better result compared to the classification of full text. On the SVM side, the classification of subjective extracts is performed slightly worse than the classification of full text. In other words, the subjectivity detection step preserves the most important information about polarity, and this aspect is captured by both classifiers. In order to double check this finding, experiments on objective extracts classification have been also performed. The objective sentences have been aggregated, building the counterparts of the subjective extracts. The micro-averaged F_1 values for the objective extracts classification were below 75% for both classifiers, hence significantly worse than both the full review and subjective extract classification. When further summarisation is performed on the subjective extracts, the results drop again. On the two sides of Table 1, we can observe a similar behaviour between summaries created from the full text and summaries created from the subjective extracts.

As further analysis, we also examine the classification of the summaries with respect to the full documents. In other words, we check if a full text and its respective summary are classified under the same label, without considering whether this is the correct answer or not. In 91% of the cases, the subjective summaries are assigned to the same label of the correspondent full text. For all the other summarisation approaches, this value drops below 80%, and in some cases below 70%. This is a further evidence of the connection between subjectivity and polarity.

5 Conclusion and Future Work

This paper has investigated the use of extractive summarisation in the context of sentiment classification. Experiments using NB and SVM classifiers have been carried out on a movie review data-set, in order to classify documents according to their polarity. Different summarisation techniques have been applied to the reviews, with the purpose of building summaries which capture the polarity of the respective original documents. Sentence extraction techniques based on statistical or positional approaches fail to capture the subjectivity of the review, and hence are inadequate to represent the sentiment of the document. On the contrary, using subjectivity detection to build subjective extracts produces results which are comparable to the full text classification. Further summarisation on top of subjectivity detection, again fail to capture the polarity of documents, as more opinion-oriented approaches needed. Showing a subjective extract instead of the full text, a potential user would only need to read 60% of a review, or even less, in order to understand its polarity.

For the future, we intend to investigate the use of knowledge extraction techniques, in order to identify entities and relationships between entities. The benefits of this approach include the opportunity of analysing opinions at a finer granularity, i.e. not only classifying the overall polarity, but also the polarity with respect to individual aspects of movies or products. This can be extended to multi-document summarisation, and would lead to the generation of personalised summaries.

References

1. P. Beineke, T. Hastie, C. Manning, and S. Vaithyanathan. Exploring sentiment summarization. In *AAAI Spring Symposium on Exploring Attitude and Affect in Text: Theories and Applications (AAAI tech report SS-04-07)*, 2004.
2. S. Blair-Goldensohn, K. Hannan, R. McDonald, T. Neylon, G.A. Reis, and J. Reynar. Building a sentiment summarizer for local service reviews. In *WWW Workshop on NLP in the Information Explosion Era*, 2008.
3. J.M. Chenlo and D.E. Losada. Effective and efficient polarity estimation in blogs based on sentence-level evidence. In *Proceedings of the 20th ACM international conference on information and knowledge management*, pages 365–374. ACM, 2011.
4. M. Hu and B. Liu. Mining opinion features in customer reviews. In *Proceedings of the National Conference on Artificial Intelligence*, pages 755–760. AAAI, 2004.
5. C.Y. Lin. Rouge: A package for automatic evaluation of summaries. In *Proceedings of the Workshop on Text Summarization Branches Out (WAS 2004)*, pages 25–26, 2004.
6. H.P. Luhn. The automatic creation of literature abstracts. *IBM Journal of research and development*, 2(2):159–165, 1958.
7. A. Nenkova and K. McKeown. Automatic summarization. *Foundations and Trends in Information Retrieval*, 5(2-3):103–233, 2011.
8. A. Nenkova and R.J. Passonneau. Evaluating content selection in summarization: The pyramid method. In *HLT-NAACL*, pages 145–152, 2004.
9. B. Pang and L. Lee. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, pages 271–278. Association for Computational Linguistics, 2004.
10. B. Pang and L. Lee. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1-2):1–135, 2008.
11. M. Potthast and S. Becker. Opinion Summarization of Web Comments. In C. Gurrin et al., editor, *Proceedings of the 32nd European Conference on Information Retrieval, ECIR 2010*, volume 5993 of *Lecture Notes in Computer Science*, pages 668–669, 2010.
12. Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM Comput. Surv.*, 34(1):1–47, 2002.
13. D. Shen, Z. Chen, Q. Yang, H.J. Zeng, B. Zhang, Y. Lu, and W.Y. Ma. Web-page classification through summarization. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 242–249. ACM, 2004.
14. P. Turney. Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL'02)*, 2002.
15. L. Zhuang, F. Jing, and X.Y. Zhu. Movie review mining and summarization. In *Proceedings of the 15th ACM international conference on Information and knowledge management*, pages 43–50. ACM, 2006.

Sull'uso di meno topic nelle iniziative di valutazione per l'information retrieval

Andrea Berto and Stefano Mizzaro

Department of Mathematics and Computer Science
University of Udine
Udine, Italy
andrea@andreaberto.it, mizzaro@uniud.it

Sommario La possibilità di ridurre il numero di topic usati in TREC e in analoghe iniziative di valutazione è stata studiata di recente, con risultati incoraggianti: anche diminuendo di molto il numero di topic (ad esempio usandone solo 10 invece di 50) è possibile, almeno potenzialmente, ottenere risultati molto simili in termini di valutazione dei sistemi. La generalità di questo approccio è però in discussione, in quanto sembra che il sottoinsieme di topic selezionato su una popolazione di sistemi sia poi non adeguato a valutare altri sistemi. In questo lavoro riconsideriamo la questione della generalità: evidenziamo alcune limitazioni dei lavori precedenti e riportiamo alcuni risultati sperimentali che sono invece più positivi. I risultati supportano l'ipotesi che con opportuni accorgimenti, i pochi topic selezionati sulla base di una popolazione di sistemi possono poi essere adeguati a valutare anche una popolazione di sistemi differente.

Keywords: TREC, valutazione, test collection, meno topic

1 Introduzione

La valutazione dei sistemi d'Information Retrieval (IR) viene spesso effettuata tramite *test collections*: questa metodologia prevede che più gruppi di ricerca partecipino ad una competizione internazionale e cerchino di reperire in modo automatico i documenti *relevant* per alcuni *topic* (ossia, descrizioni testuali di bisogni informativi). La relevance dei documenti viene decisa da giudici umani. Esistono alcune varianti di questo processo, ma le maggiori iniziative di valutazione attive oggi (TREC, NTCIR, CLEF, INEX, FIRE) lo seguono in modo abbastanza preciso.

Uno dei costi maggiori di questa metodologia è l'espressione dei giudizi di relevance, e infatti vi sono state varie proposte per cercare di diminuire questi costi [1, 2, 4, 8, 9, 11, 12, 13]. Una possibilità è quella di usare meno topic: in [3] viene evidenziato sperimentalmente che questa strada è, almeno potenzialmente, promettente; però in [7] viene invece sollevato un dubbio sulla generalità di tale risultato.

Il nostro lavoro si basa sui due lavori [3,7] appena citati. Nel paragrafo 2 i due lavori vengono descritti più in dettaglio, e ne vengono evidenziate le limitazioni

APs	t_1	$\cdots$	t_n	MAP
s_1	AP(s_1, t_1)	$\cdots$	AP(s_1, t_n)	MAP(s_1)
s_2	AP(s_2, t_1)	$\cdots$	AP(s_2, t_n)	MAP(s_2)
$\vdots$		$\ddots$		$\vdots$
s_m	AP(s_m, t_1)	$\cdots$	AP(s_m, t_n)	MAP(s_m)

Tabella 1. AP e MAP, per n topic e m sistemi (run) (da [3, pag. 21:4]).

e le domande senza risposta che motivano la necessità di continuare le ricerche in questa direzione. Nei paragrafi 3 e 4 vengono descritti alcuni ulteriori esperimenti e vengono presentati i risultati che abbiamo ottenuto, che effettivamente mitigano i problemi sulla generalità sollevati in [7].

2 I due studi

2.1 Meno topic!

Il punto di partenza del lavoro [3] è illustrato in tabella 1: ogni riga fa riferimento ad un sistema[1] ed ogni colonna ad un topic. Ogni cella della matrice AP(s_i, t_j) misura la prestazione del sistema s_i sul topic t_j; la metrica standard utilizzata in TREC è Average Precision (AP). La prestazione di un sistema s_i, solitamente, è ottenuta calcolando la media aritmetica di *tutti* i valori AP(s_i, t_j) (una riga della tabella). Questa metrica è chiamata Mean Average Precision (MAP).

Il metodo utilizzato in [3] è il seguente. Partendo dall'insieme di n topic, si considera per ogni cardinalità $c \in \{1, \ldots, n\}$ e per ogni sottoinsieme di topic di cardinalità c il corrispondente valore di MAP per ogni sistema calcolato solo su questo sottoinsieme di topic: in altri termini, si fa la media delle c (e non n) colonne in tabella 1 relative al solo sottoinsieme di topic di cardinalità c selezionato. Per ogni sottoinsieme viene poi calcolata la correlazione di questi valori di MAP con i valori di MAP dell'intero insieme di n topic. Questa correlazione misura quanto bene il sottoinsieme considerato predice le prestazioni dei sistemi in relazione all'intero insieme di topic. Per ogni cardinalità c, vengono poi selezionati i *migliori* sottoinsiemi di topic, ossia quelli con i valori di correlazione più alti. Si selezionano anche i *peggiori* sottoinsiemi e si calcola poi la correlazione *media* su tutti i sottoinsiemi di cardinalità c.

In [3] vengono usati dati di TREC 8 [10] (da cui sono stati eliminati il 25% dei sistemi peggiori: tabella 1 con $n = 50$ e $m = 96$) ed NTCIR 6 (tabella 1 con $n = 50$ e $m = 74 - 25\% = 56$), varie metriche di efficacia (oltre a MAP, anche RPrec, P@10, GMAP, ed NDCG) e varie misure di bontà dei sottoinsiemi di topic (oltre alla Correlazione, anche Tau di Kendall e Tasso d'errore).

Il grafico in figura 1 riassume il risultato principale: i valori di correlazione per ogni cardinalità. Esso mostra che il miglior sottoinsieme di cardinalità, ad

[1] Anche se sarebbe più corretto, in terminologia TREC, usare *run*.

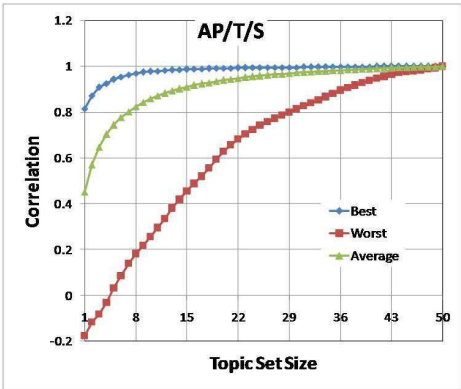


Figura 1. Correlazioni massima, media e minima per cardinalità. Misura MAP (da [3, pag. 21:5]).

esempio, $c = 5$ o $c = 10$ è decisamente migliore nel prevedere le prestazioni sull'intero insieme di 50 topic rispetto ad un sottoinsieme di pari cardinalità scelto a caso, il quale a sua volta si comporta molto meglio del peggior sottoinsieme. Interpretando la figura orizzontalmente, se l'obiettivo è una correlazione di 0.95 rispetto all'intero insieme, la scelta del miglior sottoinsieme permette di poter utilizzare soltanto 6 topic, rispetto ai 22 necessari se si sceglie un sottoinsieme casuale ed ai 41 se la scelta ricade sul peggior sottoinsieme. Risultati simili vengono riportati per le altre metriche di efficacia e misure di bontà.

In [3] sono studiati anche altri sottoinsiemi di topic con buona correlazione, i cosiddetti "best set": analizzando i 10 migliori sottoinsiemi per ogni cardinalità c, risulta che questi sono abbastanza differenti fra di loro. Inoltre viene analizzato anche il problema della generalizzazione, ossia di quanto i sottoinsiemi di buoni topic trovati sulla base di una certa popolazione di sistemi risultino buoni topic anche quando si misurano le prestazioni di un'altra popolazione di sistemi. Questo studio viene effettuato spezzando in due la popolazione dei sistemi partecipanti a TREC 8, ma lascia il dubbio che i run multipli effettuati con un unico sistema inficino in qualche modo l'esperimento.

2.2 Meno topic?

In [7] la generalizzazione viene ulteriormente studiata. Per fare ciò, oltre ai dati sui 96 sistemi di TREC 8 usati in [3] (denominati TREC96), vengono usate due nuove popolazioni di sistemi: TREC87 (TREC 8 senza i sistemi manual, per avere una popolazione di sistemi più omogenea) e Terrier (20 run di differenti varianti del sistema Terrier [5, 6]) per avere una popolazione di sistemi completamente differente seppure sugli stessi topic.

L'obiettivo principale di [7] è di capire se i migliori sottoinsiemi di topic selezionati per le varie cardinalità $c \in \{1, \ldots, n\}$ su una popolazione di sistemi

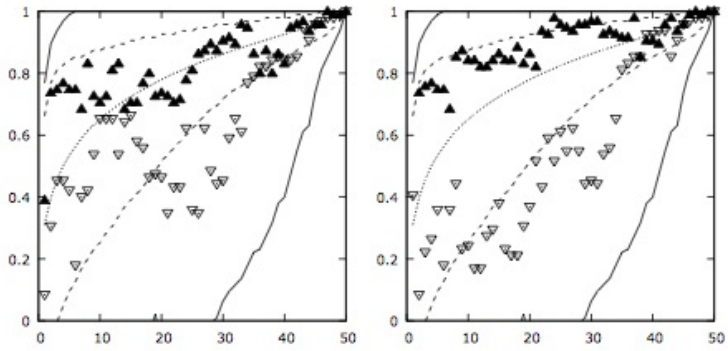


Figura 2. Tau di Kendall del migliore sottoinsieme di TREC96 (sinistra) e TREC87 (destra) applicati su Terrier (da [7, pag. 138]).

(vengono usate TREC96 e TREC87) risultano essere dei buoni sottoinsiemi di topic anche per valutare un'altra popolazione di sistemi (Terrier).

La figura 2 mostra il risultato ottenuto. Le cinque linee rappresentano rispettivamente i valori di correlazione massimi, il 95esimo percentile, medi (ossia, quelli attesi selezionando un sottoinsieme casuale di topic), il 5o percentile e peggiori ottenuti per Terrier; i triangoli pieni con punta verso l'alto sono i valori di correlazione dei sottoinsiemi migliori, ricavati su TREC96 o TREC87 e applicati a Terrier. Il risultato è piuttosto negativo, soprattutto per TREC96: il miglior sottoinsieme di topic, per ciascuna cardinalità, tende a comportarsi sempre meno bene del 95esimo percentile, e spesso anche peggio di un sottoinsieme di topic casuale della stessa cardinalità. I migliori sottoinsiemi selezionati su TREC87 sembrano comportarsi meglio: quando usati su Terrier portano a correlazioni vicine al 95esimo percentile e quasi sempre hanno una correlazione maggiore di un sottoinsieme casuale di topic.

2.3 Limitazioni e motivazioni

Il lavoro [7] mette quindi in discussione il risultato almeno potenzialmente positivo di [3]: sembra che i sottoinsiemi di topic adeguati per valutare una popolazione di sistemi non siano poi adeguati per valutare una popolazione di sistemi differente. Si possono però evidenziare alcune alcune limitazioni:

- L'analisi viene effettuata usando solo il singolo "best set"; resta in dubbio se vi siano altri sottoinsiemi di topic che siano buoni quasi quanto il migliore sottoinsieme di topic sulla popolazione di partenza, e che altresí generalizzino bene, ossia presentino una buona correlazione anche su una popolazione di sistemi differente.
- Vengono usate soltanto Tau di Kendall (non le altre misure di bontà) e GMAP e logit(AP) (e non le altre metriche di efficacia). I risultati potrebbero essere differenti per altre combinazioni di misure/metriche.

– Inoltre in nessuno dei due lavori [3, 7] viene detto nulla sul *numero* di best set: ossia, non è chiaro se vi siano molti o pochi sottoinsiemi di topic buoni (che consentono di valutare essenzialmente in modo analogo i sistemi).

Ha quindi senso continuare questa linea di ricerca. In questo lavoro ci chiediamo:

D1. Quanti "best set" ci sono?
D2. Se invece di considerare il singolo "best set" come fatto in [7] se ne considerano di più, i risultati sulla generalizzazione sono più positivi? In altri termini, se si considerano i 10 best set, quanto questi sono generali?

3 Esperimento 1: quanti "good subset"?

Per poter rispondere a **D1**, ossia sapere quanti "good subset" esistono, è stato condotto l'esperimento seguente. Per ogni cardinalità abbiamo usato l'euristica presentata in [3] per selezionare 10 milioni di sottoinsiemi di topic[2] e per ognuno di essi è stata calcolata la MAP parziale e la correlazione lineare di quest'ultima con la MAP dell'intero insieme di topic. Considerando 0.96 come soglia di correlazione oltre la quale un sottoinsieme predice bene i risultati finali, abbiamo contato il numero di sottoinsiemi che superano tale soglia. L'esperimento è stato condotto su tutte e tre le collezioni (TREC96, TREC87, Terrier) e abbiamo preso in esame, oltre alla correlazione, anche la Tau di Kendall (con soglia 0.85 anziché 0.96).

La figura 3 riporta i risultati sulle collezioni TREC96, TREC87 e Terrier. Essa mostra, per ogni cardinalità di ogni collezione, il numero di sottoinsiemi, tra i 10 milioni considerati, che hanno un valore di correlazione superiore a 0.96 e di Tau superiore a 0.85. Analizziamo prima le curve relative alla correlazione. Per quanto riguarda TREC96, si nota come il numero di "good subset" cresca velocemente: a cardinalità 25, ad esempio, più della metà dei sottoinsiemi calcolati è costituita da buoni sottoinsiemi e dalla cardinalità 35 si supera il 99% di "good subset".

In TREC87 le quantità di "good subset" sono simili anche se leggermente inferiori; questo è probabilmente dovuto all'assenza dei run manuali, notoriamente più efficaci e tali da esercitare una forte influenza nel calcolo dei risultati finali. Per Terrier i valori sono invece leggermente superiori, specie a cardinalità basse, dove si registrano già numerosi buoni sottoinsiemi (ad esempio a cardinalità 9 oltre il 20% dei sottoinsiemi risulta essere un "good subset").

Considerando la Tau di Kendall, i risultati ottenuti sono leggermente più bassi per ognuna delle tre collezioni analizzate: TREC96 riporta comunque un numero di "good subset" maggiore di TREC87 (i run manuali sono influenti indifferentemente dalla misura considerata), ma minore di Terrier (collezione formata da pochi run e molto simili tra loro).

L'esistenza di un numero così alto di sottoinsiemi di topic con buona correlazione fa pensare che sia effettivamente possibile trovarne di generali. Per studiare questo aspetto abbiamo eseguito un secondo esperimento.

[2] Per le cardinalità da 1 a 5 si sono analizzati tutti i possibili sottoinsiemi.

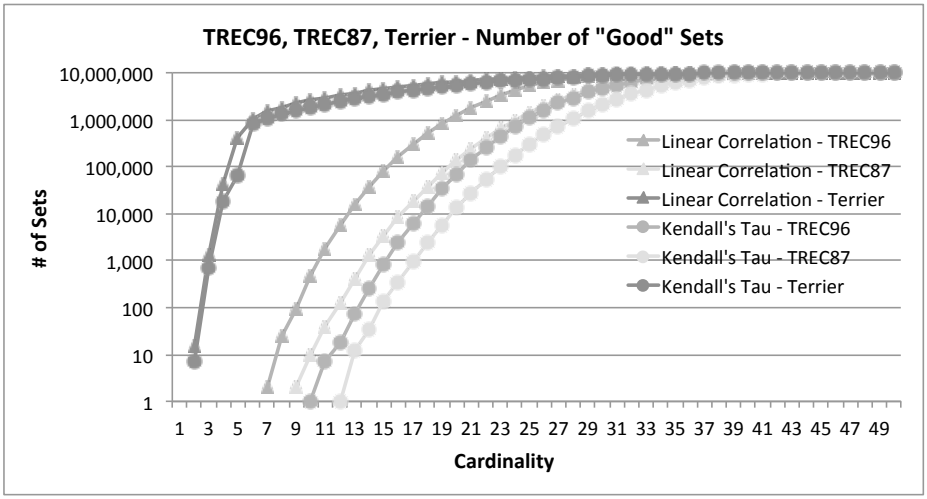


Figura 3. Il numero di "buoni" sottoinsiemi di topic alle varie cardinalità (scala semilogaritmica) per le 3 collezioni.

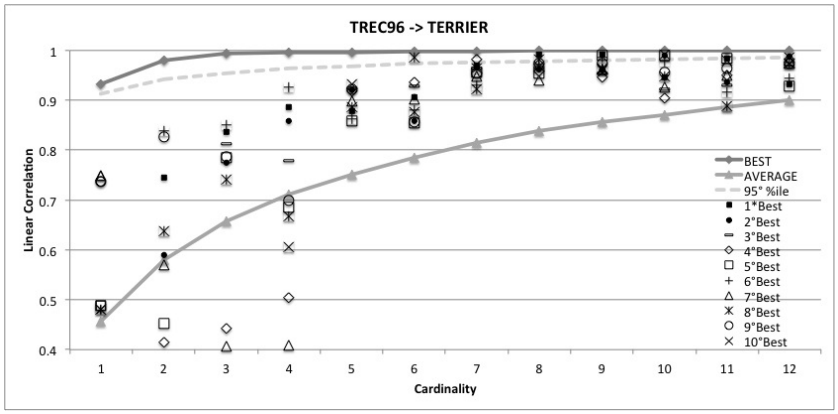
4 Esperimento 2: generalizzazione

Per poter rispondere alla seconda domanda **D2** è stato condotto un esperimento di generalizzazione prendendo da TREC96 e TREC87, per ogni cardinalità, i migliori 10 sottoinsiemi di topic e usandoli per valutare Terrier. L'obiettivo è di capire se fra i 10 migliori sottoinsiemi di topic ce ne sono alcuni che generalizzano (mentre in [7] si è guardato solo il migliore). In questo modo viene effettuato un test di generalità sulla capacità di valutazione dei migliori sottoinsiemi su una collezione diversa da quella da cui sono ricavati.

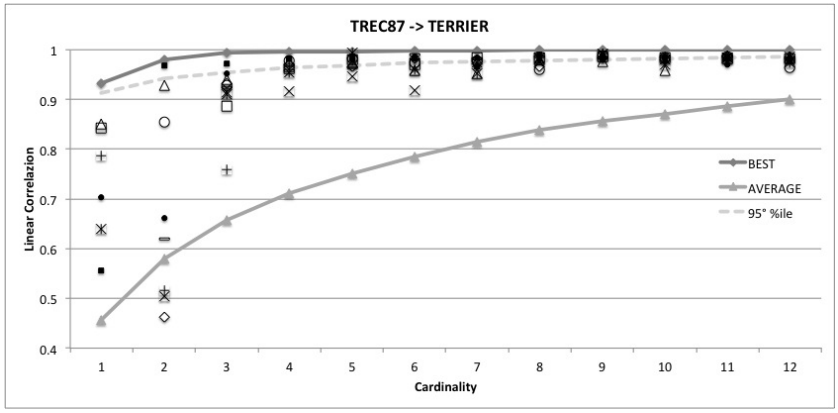
L'esperimento è stato svolto, finora, per le cardinalità da 1 a 12 (l'alto numero di sottoinsiemi rende il problema computazionalmente pesante, come discusso in [3]). Per ognuno dei 10 migliori sottoinsiemi ottenuti su TREC96 e TREC87 e per ogni cardinalità è stata calcolata la MAP parziale sui sistemi della collezione Terrier; questo valore è poi stato correlato, mediante sia la correlazione sia la Tau di Kendall, con la MAP totale sui sistemi della collezione Terrier. In questo modo si sono ottenuti 10 valori di correlazione per ognuna delle 12 cardinalità, riferiti ai migliori sottoinsiemi calcolati su TREC96/87 e generalizzati su Terrier.

Le figure 4 e 5 riportano i risultati, rispettivamente in termini di correlazione e Tau. Nelle figure, le tre linee rappresentano i valori di correlazione massimi, il 95esimo percentile e medi: sono analoghe alle tre linee più in alto di figura 2 (sono differenti perché qui è stata usata la metrica MAP anziché logit(AP)). I punti rappresentano i valori di correlazione per i 10 best set ottenuti su una popolazione di sistemi differente.

Si può notare come la maggior parte dei punti in figura 4 stia al di sopra della linea media; per TREC87 molti sono anche al di sopra del 95esimo percentile. Questo risultato è più positivo di quello ottenuto in [7]: se si considerano i 10



(a)



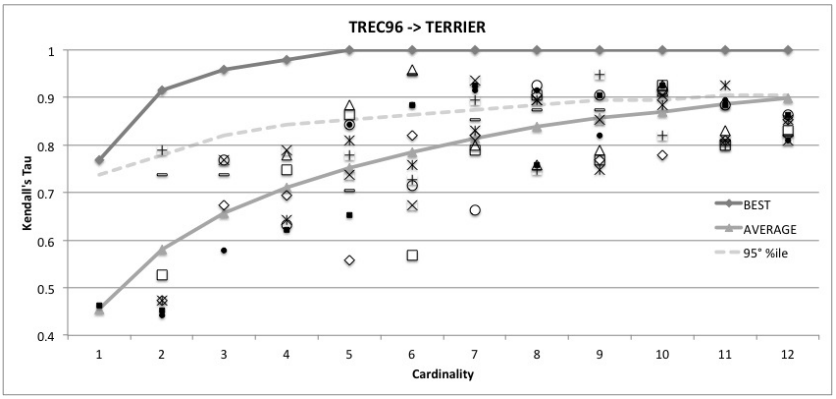
(b)

Figura 4. Generalizzazione: correlazioni dei 10 best set secondo TREC96 (a) e TREC87 (b) su Terrier.

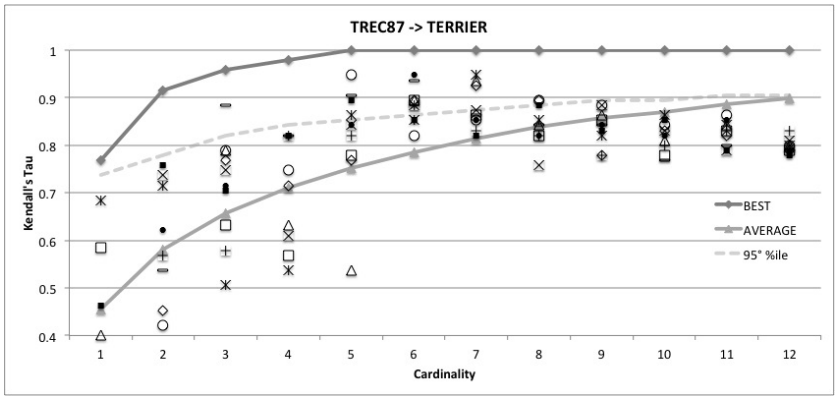
migliori sottoinsiemi di topic ottenuti sulla base di una certa popolazione di sistemi, fra di essi molti sono adeguati a misurare le prestazioni anche di altre popolazioni di sistemi. Il fatto che TREC87 si comporti sistematicamente meglio di TREC96 inoltre è positivo, in quanto lascia intravedere un modo di scegliere la popolazione di sistemi in cui cercare i sottoinsiemi di topic generali (è meglio se è omogenea).

Tau di Kendall presenta risultati un po' più negativi della correlazione lineare: in figura 5 molti punti sono al di sotto non solo del 95esimo percentile ma anche della linea mediana. Questo significa che i best set sono più efficaci nel predire il valore di MAP che nell'ordinare i sistemi allo stesso modo dell'insieme di tutti e 50 i topic.

Viene spontaneo a questo punto porsi una terza domanda:



(a)



(b)

Figura 5. Generalizzazione: Tau di Kendall dei 10 best set secondo TREC96 (a) e TREC87 (b) su Terrier.

D3. L'ordine dei "best set" si ripercuote sulla capacità di generalizzazione dei sottoinsiemi? Ossia: il primo best set tende ad essere migliore (quando usato su una popolazione di sistemi differente) del secondo, e questo a suo volta tende ad essere migliore del terzo e così via?

Una prima risposta negativa viene già dal risultato di [7], ma si può essere più sistematici ed analizzare tutti i migliori 10 best set. Le figure 4 e 5 non consentono di rispondere, e quindi nelle figure 6 e 7 vengono riportati gli stessi risultati (i valori di correlazione e Tau al variare delle cardinalità per i 10 best set) in una forma grafica più appropriata. Dall'andamento ondulato (più evidente per la Kendall di Tau, in figura 7) è chiaro che la risposta a **D3** è negativa. Quindi per trovare il sottoinsieme di topic che generalizza meglio non ci si può basare solo sulla bontà di tale sottoinsieme sulla popolazione di partenza, ma bisogna

considerare vari sottoinsiemi.

5 Conclusioni e sviluppi futuri

In questo lavoro abbiamo rivisto ed esteso alcuni risultati ottenuti in [3, 7]. Sulla base degli esperimenti effettuati, e ancora in corso, sembra che:

- se si cerca di predire le prestazioni di una popolazione di sistemi usando un sottoinsieme di topic di cardinalità ridotta rispetto agli usuali 50 topic di TREC, esistono *molti* sottoinsiemi di topic "buoni";
- se si selezionano i sottoinsiemi di topic "buoni" su una popolazione di sistemi, anche se il migliore di tali sottoinsiemi per ogni cardinalità sembra non essere generale (ossia, sembra non adeguato a valutare le prestazioni su un'altra popolazione di sistemi [7]), in realtà la situazione migliora se si considerano i successivi "buoni" sottoinsiemi: molti fra questi sono invece adeguati.

Gli esperimenti di generalizzazione presentati in questo lavoro riguardano soltanto le cardinalità da 1 a 12 e prendono in considerazione solamente la metrica MAP. Questa limitazione è dovuta alla complessità computazionale nel calcolo di tutti i possibili sottoinsiemi a cardinalità maggiore di 12 (e, specularmente, minore di 38), soprattutto per quanto riguarda la Tau di Kendall. Per poter confrontare in maniera più diretta i risultati ottenuti con i risultati presentati in [7], è in corso di elaborazione un esperimento che utilizza come metrica logit(AP), la stessa di [7], invece di MAP.

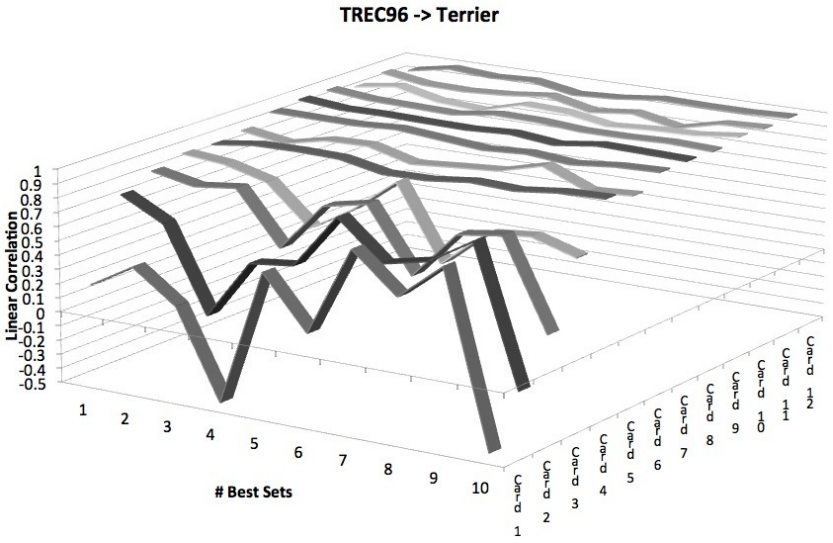
Un'altra possibile estensione del lavoro riguarda lo studio della generalizzazione per le cardinalità da 38 a 50 (i cui dati sono calcolabili in tempi accettabili). Tuttavia, le cardinalità di maggior interesse per lo scopo che si prefigge lo studio (la sensibile riduzione del numero di topics), sono probabilmente quelle comprese tra circa 5 e circa 20, ragionevolmente coperte dal lavoro presentato. Inoltre, come fatto già in [3], sarà importante verificare i risultati, oltre che sui dati di TREC, anche sui dati delle altre iniziative di valutazione.

Ringraziamenti

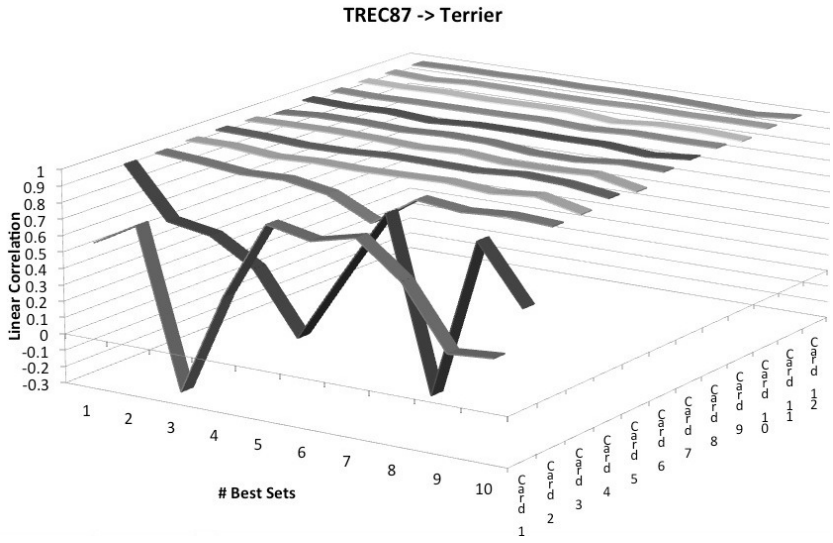
Ringraziamo Steve Robertson per aver fornito alcuni dati per gli esperimenti e per alcuni utili suggerimenti.

Riferimenti bibliografici

1. C. Buckley and E. Voorhees. Evaluating evaluation measure stability. In N. J. Belkin, P. Ingwersen, and M.-K. Leong, editors, *SIGIR 2000: Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 33–40, New York, 2000. ACM Press.

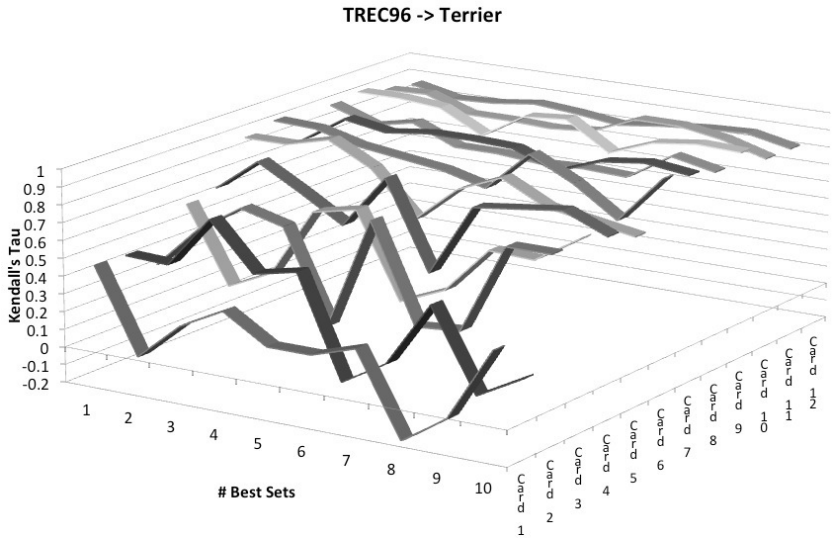


(a)

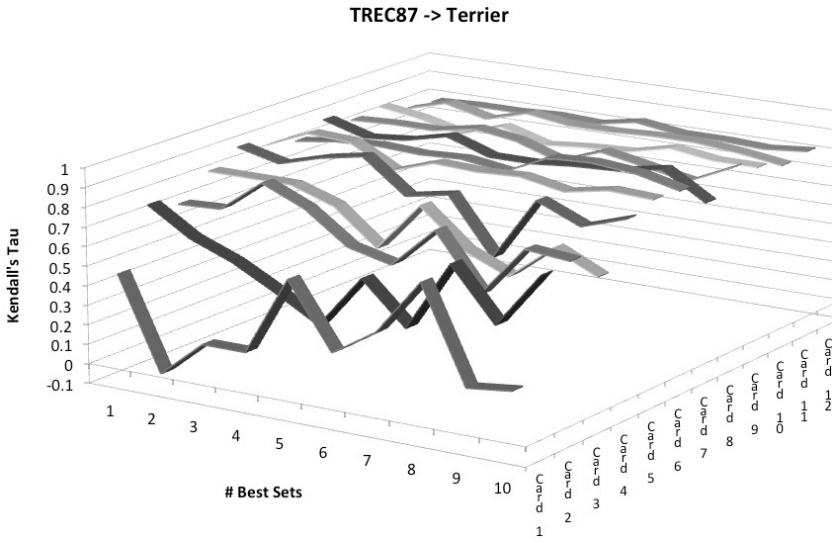


(b)

Figura 6. Andamento della correlazione dei 10 best set secondo TREC96 (a) e TREC87 (b) su Terrier.



(a)



(b)

Figura 7. Andamento della Tau di Kendall dei 10 best set secondo TREC96 (a) e TREC87 (b) su Terrier.

2. B. Carterette, J. Allan, and R. Sitaraman. Minimal test collections for retrieval evaluation. In E. N. Efthimiadis, S. T. Dumais, D. Hawking, and K. Järvelin, editors, *SIGIR 2006: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 268–275, New York, 2006. ACM Press.
3. J. Guiver, S. Mizzaro, and S. Robertson. A few good topics: Experiments in topic set reduction for retrieval evaluation. *ACM Transactions on Information Systems (TOIS)*, 27(4), November 2009.
4. S. Mizzaro and S. Robertson. HITS hits TREC — exploring IR evaluation results with network analysis. In C. L. A. Clarke, N. Fuhr, N. Kando, W. Kraaij, and A. P. de Vries, editors, *SIGIR 2007: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 479–486, New York, 2007. ACM Press.
5. I. Ounis, G. Amati, P. V., B. He, C. Macdonald, and Johnson. Terrier Information Retrieval Platform. In *Proceedings of the 27th European Conference on IR Research (ECIR 2005)*, volume 3408 of *LNCS*, pages 517–519. Springer, 2005.
6. I. Ounis, C. Lioma, C. Macdonald, and V. Plachouras. Research directions in terrier. *Novatica/UPGRADE Special Issue on Web Information Access, Ricardo Baeza-Yates et al. (Eds), Invited Paper*, 2007.
7. S. Robertson. On the Contributions of Topics to System Evaluation. In *Advances in Information Retrieval*, volume 6611 of *Lecture Notes in Computer Science*, pages 129–140. Springer Berlin / Heidelberg, 2011.
8. M. Sanderson and J. Zobel. Information retrieval system evaluation: effort, sensitivity, and reliability. In G. Marchionini, A. Moffat, J. Tait, R. Baeza-Yates, and N. Ziviani, editors, *SIGIR 2005: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 162–169, New York, 2005. ACM Press.
9. E. Voorhees and C. Buckley. The effect of topic set size on retrieval experiment error. In M. Beaulieu, R. Baeza-Yates, S. H. Myaeng, and K. Jarvelin, editors, *SIGIR 2002: Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 316–323, New York, 2002. ACM Press.
10. E. M. Voorhees and D. Harman. Overview of the Eighth Text REtrieval Conference (TREC-8). In *TREC*, 1999.
11. W. Webber, A. Moffat, and J. Zobel. Statistical power in retrieval experimentation. In *CIKM '08: Proceeding of the 17th ACM conference on Information and knowledge management*, pages 571–580, New York, NY, USA, 2008b. ACM.
12. E. Yilmaz and J. A. Aslam. Estimating average precision with incomplete and imperfect judgements. In P. S. Yu, V. J. Tsotras, E. A. Fox, and B. Liu, editors, *CIKM 2006: Proceedings of the 13th ACM Conference on Information and Knowledge Management*, pages 102–111, New York, 2006. ACM Press.
13. J. Zobel. How reliable are the results of large-scale information retrieval experiments? In W. B. Croft, A. Moffat, C. J. van Rijsbergen, R. Wilkinson, and J. Zobel, editors, *SIGIR'98: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 307–314, New York, 1998. ACM Press.

Classical vs. Crowdsourcing Surveys for Eliciting Geographic Relevance Criteria

Stefano De Sabbata[1], Omar Alonso[2], and Stefano Mizzaro[3]

[1] University of Zurich-Irchel
Winterthurerstrasse 190, CH-8057 Zurich, Switzerland
stefano.desabbata@geo.uzh.ch
[2] Microsoft Corp.
1065 La Avenida, Mountain View CA, USA
omar.alonso@microsoft.com
[3] University of Udine
Via delle Scienze 206, 33100 Udine, Italy
mizzaro@uniud.it

Abstract. Geographic relevance aims to assess the relevance of physical entities (e.g., shops and museums) in geographic space for a mobile user in a given context, thereby shifting the focus from the digital world (the realm of classical information retrieval) to the physical world. We study the elicitation of geographic relevance criteria by means of both a classical survey and an Amazon Mechanical Turk (a crowdsourcing platform) survey. This allows us to obtain three results: first, we gather a set of criteria and their relative importance; second, we gain a first insight on the differences between geographic relevance and classical relevance as commonly understoon in the IR field; and third we draw some considerations on the agreement, on the importance of specific criteria, among the participants to the classical and the crowdsourcing surveys.

Keywords: Relevance, Crowdsourcing, Amazon Mechanical Turk, SurveyMonkey

1 Introduction

The elicitation of relevance criteria dates back to the 90s, if not earlier [7]. Although such criteria seemed quite well established at that time [2], recently this issue is studied again [1]. This is probably due to the Web, that on the one side provides novel search services that might entail a different notion of relevance, and on the other side allows more convenient methods for preparing surveys involving several participants.

In this short paper, we concentrate on Geographic Relevance (GR), a recent area of Information Retrieval (IR), and we discuss the elicitation of relevance criteria by means of:

– SurveyMonkey (SM, www.surveymonkey.com), a Web service that allows the preparation of an online survey whose participants are then invited by email, and
– Amazon Mechanical Turk (AMT, www.mturk.com), a crowdsourcing platform that allows to outsource to the crowd specific tasks for a small amount of money.

The aim of this research is threefold:

– to find suitable GR criteria, that might be different from the classical relevance criteria;
– to gain a first insight into the difference between GR and the classical concept of relevance in the IR field;
– to understand if AMT provides reliable results, or at least if those results agree with the SM ones, which are obtained in a more classical way.

AMT quality and reliability are important issues [6]: there is no guarantee that AMT workers provide reliable answers and that they carry on their task in a reliable way; for example, workers might cheat to quickly gain money. This is even more critical as crowdsourcing is emerging as a widespread alternative for relevance evaluations.

In the following, we first define GR (Section 2) and discuss crowdsourcing and AMT (Section 3) then we present the experimental study and its results (Section 4), and we finally summarize the main findings (Section 5).

2 Geographic Relevance Criteria

The basic idea of GR is to assess the relevance of *physical entities* (e.g., shops and museums) in geographic space for a mobile user in a given context [8]. This definition implies a shift from the informational world — that is the focus of IR, which is devoted to retrieve information from unstructured digital document collections — to the physical world. In other terms, the aim of GR is to apply the principles and concepts developed in the field of IR not only in the informational world, but also in the physical world [3].

GR is different from Geographic Information Retrieval because the second still focuses on digital entities. The aim of Geographic Information Retrieval is to retrieve geographic information from digital documents, or to find relevant digital documents that can satisfy a user's need for geographic information. GR uses digital entities (e.g., the objects in a collection within a Geographic Information System, or documents, or images, etc.) as means to estimate the relevance of the physical entities they refer to, rather than aiming to evaluate the relevance of the digital entities themselves.

In shifting the focus from the digital world to the physical world, a first question is whether the criteria of relevance developed in IR [7, 2, 1] can be applied to assess GR. A second question is whether other criteria are needed in order to fully understand the relevance of a physical entity. We ground our study

Properties	**Geography**	**Information**	**Presentation**
Topicality	Spatial proximity	Specificity	Accessibility
Appropriateness	Temporal proximity	*Availability*	Clarity
Coverage	*Spatio-temporal proximity*	*Accuracy*	Tangibility
Novelty	*Directionality*	*Currency*	*Dynamism*
	Visibility	Reliability	*Presentation quality*
	Hierarchy	Verification	
	Cluster	Affectiveness	
	Co-location	Curiosity	
	Association rule	Familiarity	
		Variety	

Table 1. Four sets of GR criteria, classified as in [4].

on the set of criteria of GR proposed in [4]; these criteria are listed in Table 1. We do not have the space here to discuss these criteria in detail; a comprehensive description of each single criterion, together with a more in depth analysis, is provided in [5].

3 Crowdsourcing

Crowdsourcing has emerged as a feasible alternative for relevance evaluation because it brings the flexibility of the editorial approach at a larger scale.

AMT is an example of a crowdsourcing platform: it is an Internet service that gives developers the ability to include human intelligence as a core component of their applications. Developers use a web services API to submit tasks, approve completed tasks, and incorporate the answers into their software applications. To the application, the transaction looks very much like any remote procedure call: the application sends the request, and the service returns the results. People (the "crowd") come to the web site looking for tasks and receive payment for their completed work. In addition to the API, there is also the option to interact using a dashboard that includes several useful features for prototyping experiments. There is an increased participation by large numbers of online users from all over the world, which is a good sample that includes diversity.

The individual or organization who has work to be performed is known as the *requester*. A person who wants to sign up to perform work is described in the system as a *worker*.

One issue with AMT and similar crowdsourcing platform is quality [6]: there is no guarantee that the workers provide correct answers and that they carry on their task in a reliable way. For example, workers might cheat to quickly gain money. One of the aims of this paper is to compare a survey carried on by means of AMT with a similar one carried on by more classical means, like SM.

1. Considering a place that fits your needs by its category (e.g. a restaurant, if you want to go out for dinner), which other criteria would you take into account?
   - A place that offers **just** the services you need is more relevant than a place that also offers **other** services.
   - A place that offers **all** the services you need is more relevant than a place that offers just **some** of them.
   - A place that was **previously unknown** to you is more relevant than an **already known** place.
2. Considering a place that fits your needs, do you take into account the following criteria related to the presented information and the way it is presented (for example on your mobile device) to judge its relevance?
   - The more information available about a place, the higher is the relevance of the place.
   - The more accurate the information about a place, the higher is the relevance of the place.
   - The more current, recent, timely, up-to-date the information about a place, the higher is the relevance of the place.
   - The more dynamic, active or interactive the presentation of information, the higher is the relevance of the presented place.
   - The more the information about a place is presented in a certain format or style, or offers output in a way that is helpful, desirable, or preferable, the higher is its relevance.

Fig. 1. Questions 1 and 2 as framed in SMs and AMTs1.

4 Experiments

4.1 Experimental design

We selected a subset of the criteria listed in Table 1: the 14 criteria in italics. We chose many of the geographic criteria, leaving out *spatial proximity* and *temporal proximity* (we took into account the *spatio-temporal proximity* that combines both), and *association rule* (which is difficult to explain and can be misunderstood if not explained in detail). We selected two or three criteria from each of the other groups, choosing the easier to explain in a few words and, probably, the most intuitive ones.

Towards the aims stated in Section 1, we ran 3 experiments:

- A SM survey (referred to as SMs) sent by email to researchers and students in IR and similar subjects.
- A first AMT survey (AMTs1) obtained by simplifying the SM survey and by focussing on some items only.
- A second AMT survey (AMTs2) obtained, after the responses to AMTs1, by fine tuning the language to tailor it to the AMT environment, where workers usually are not keen to spend much time on a task.

The questions were asked in an indirect way: for example, we did not ask literally whether "*spatio-temporal proximity* is an important GR criterion"; rather

1. Given a place in the right category (e.g., a restaurant, if you want to go out for dinner), which other criteria would you take into account?
   - A place that offers **just** the services you need is more relevant than a place that also provides **other** services.
   - A place that offers **all** the services you need is more relevant than a place that provides just **some** of them.
   - A place that was **previously unknown** to you is more relevant than an **already known** place.
2. Considering a place that fits your needs, do you take into account the following criteria to judge its relevance?
   - The **more information** available about a place, the higher is the relevance of the place.
   - The more **accurate** the information about a place, the higher is the relevance of the place.
   - The more **current, recent, timely, up-to-date** the information about a place, the higher is the relevance of the place.
   - The more **dynamic, active or interactive the presentation** of information, the higher is the relevance of the presented place.
   - The more the information about a place is presented in a certain **format or style, or offers output in a way that is helpful, desirable, or preferable**, the higher is its relevance.

Fig. 2. Questions 1 and 2 as framed in AMTs2.

we asked whether "it is important to take into account whether the place (or a related event) will be available at the time you will be able to reach it (e.g., whether you can reach the shop before it closes)." The questionnaire included a total of 14 items, arranged into three main questions.

Figure 1 shows two of the three questions (each one grouping some items) as framed in SMs and AMTs1. In SMs, a first page was dedicated to the criteria not related to geographic concepts (e.g., *novelty*), whereas a second page was dedicated to the geography-related criteria. The same items have been used in AMTs1, where the 3 questions were all presented in one page. Figure 2 shows the same items as framed in AMTs2, where we slightly modified the questions (but not the items, that were almost identical to SMs and ATMs1[4]), each one presented in a separate page. Participants assessed each item on a 7-point Likert scale "1 - Strongly disagree" – "7 - Strongly agree" (all the scale values appear on the ordinal axis in Figure 3).

4.2 Results

The number of participants in the three cases is similar: SMs got 53 participants, AMTs1 43, and AMTs2 42 (we discarded two outliers from each AMT survey since they were far too quick). The collected demographics say that participants

[4] The only differences, as shown in the figures, is the change of "offer" into "provide" and the usage of boldface to highlight some terms.

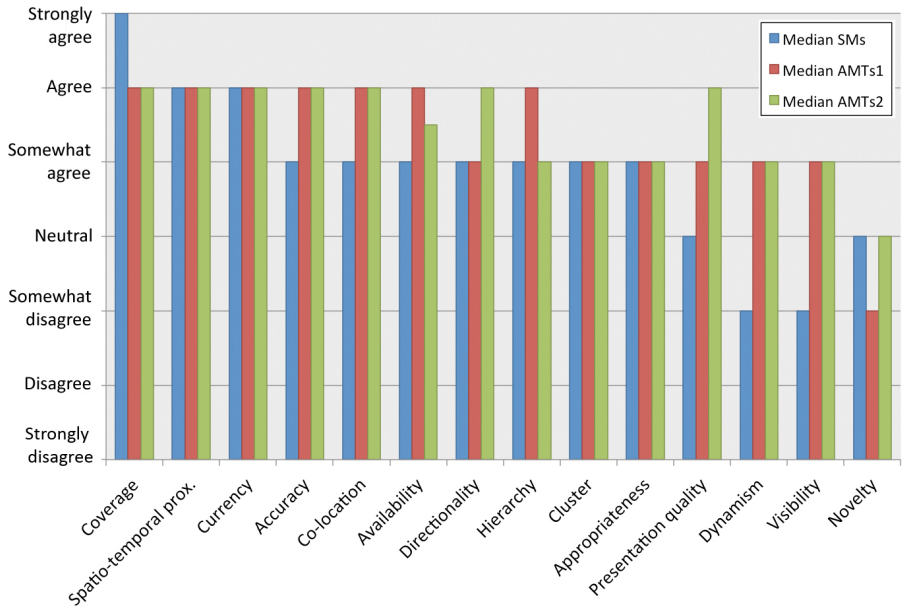


Fig. 3. Median value for each criteria.

to SMs were familiar with digital maps (71% use them at least several times a week), mobile maps (51% use them on their mobile), and online yellow pages (only 30% of the participants have never used them). We did not collect demographic data for AMT (we plan to do that in future experiments). We paid \$0.15 to each AMT worker. The total cost for both AMT experiments was \$16.

The Kolmogorov-Smirnov normality test was negative, so we considered the variables as ordinal. Figure 3 shows the median importance of the single criteria in the three surveys.

By analyzing the relative importance of the criteria, three groups can be singled out: a first one including the three leftmost criteria (*coverage*, *spatio-temporal proximity*, and *currency*), whose importance seems very high according to all the three surveys; a second group including the central seven criteria whose importance is tangible, but somehow lower with respect to the first group; and a final group of the four rightmost criteria whose importance seems rather low and more inconsistent among the three surveys.

Turning to the agreement among the participants in the three surveys, we can note first that SMs median values are generally lower than AMTs1/2. Also, agreement is different for each criterion, as confirmed by a Mann-Whitney test:

– highly significant ($p < .01$) difference has been found between SMs and AMTs1, and also between SMs and AMTs2, for the criteria *availability*, *accuracy*, *dynamism*, *presentation quality*;

	SM	**AMT**
Demographics	Targeted practitioners and experts	Crowd (unknown workers)
Incentive	Volunteer	Money
Development cost	low	low
Service fee	$30 per month	Free
Participant fee	None	$0.15 per participant
Cost dependencies	Time and service level	Number of participants per survey
Total incurred cost	$60	$8 + $8
Time to completion	45 days	3 days for AMTs1 and 6 days for AMTs2

Table 2. SM vs. AMT comparison.

- highly significant ($p < .01$) difference has been found between SMs and AMTs1 for the criterion *hierarchy*, and between SMs and AMTs2 for the criterion *visibility*;
- significant ($p < .05$) difference has been found between SMs and AMTs1 for the criteria *currency* and *visibility*, and between SMs and AMTs2 for the criterion *co-location*;
- no statistical significant difference has been found between AMTs1 and AMTs2, in any criteria.

Besides differences in quality per se, there are other characteristics that may influence the choice of system for conducting surveys. We present the most important aspects in Table 2.

5 Conclusions

Overall, the results hint that:

- The most important GR criteria seem to be *coverage*, *spatio-temporal proximity*, and *currency*.
- SM and AMT surveys provide slightly different results.
- The differences mainly concern the importance of four criteria (*availability*, *accuracy*, *dynamism* and *presentation quality*)
- None of these four criteria are in the *Geography* set (see Table 1).

This last point is perhaps surprising, since one would expect that the heterogeneous background and cultural differences of the international AMT population would particularly affect the elicitation of geographic criteria. However, in our experiments disagreement was mainly on classical relevance criteria.

One further point to remark is that the average quality of AMT workers answers was good, as demonstrated by the good agreement level with SM, although we did not require qualified workers — as it would have been possible in AMT.

Finally, as future work, we are considering a more "visual" survey, with more images or scenarios, than just pure text as we did in this work .

References

1. O. Alonso and S. Mizzaro. Relevance criteria for e-commerce: a crowdsourcing-based experimental analysis. In *SIGIR '09: Proceedings of the 32nd international ACM SIGIR*, pages 760–761, 2009.
2. C. L. Barry and L. Schamber. Users' criteria for relevance evaluation: A cross-situational comparison. *Information Processing & Management*, 34(2-3):219–236, May 1998.
3. P. Coppola, V. D. Mea, L. D. Gaspero, and S. Mizzaro. The concept of relevance in mobile and ubiquitous information access. In *Mobile HCI Workshop on Mobile and Ubiquitous Information Access*, volume 2954 of *LNCS*, pages 1–10. Springer, 2003.
4. S. De Sabbata. Criteria of geographic relevance. In *6th Int'l Conf. on Geographic Information Science*, 2010.
5. S. De Sabbata and T. Reichenbacher. Criteria of geographic relevance: an experimental study. *International Journal of Geographic Information Science*, forthcoming.
6. P. Marsden. Crowdsourcing. *Contagious Magazine*, 18:24–28, 2009.
7. S. Mizzaro. Relevance: The whole history. *Journal of the American Society for Information Science*, 48(9):810–832, 1997.
8. T. Reichenbacher, P. Crease, and S. De Sabbata. The concept of geographic relevance. In *Proceedings of the 6th Int'l Symposium on LBS & TeleCartography*, 2009.

Flexible Querying in Geo-Finder

Gloria Bordogna[1], Giuseppe Psaila[2]

[1] CNR-IDPA - via Pasubio 5, I-24044 Dalmine (BG) (Italy)
gloria.bordogna@idpa.cnr.it
[2] Università di Bergamo - viale Marconi 5, I-24044 Dalmine (BG) (Italy)
psaila@unibg.it

Abstract. The evaluation of queries specifying both content based conditions and spatial conditions on documents contents in Geographic Information Retrieval requires representing the vagueness and context dependency of spatial conditions and the personal user's preferences.
The *Geo-Finder* system [1] implements a Geo-Retrieval model that evaluates flexible spatial queries combined with content queries. The spatial condition is interpreted as the soft constraint "close" on the user's perceived distance. Two distinct semantics can be used to combine the spatial and the content conditions: *and possibly* or *average*; in both cases it is possible to modify the relative weight (preference) of conditions.

Keywords: Geographic Information Retrieval, Fuzzy aggregation operators, context dependent spatial query, soft constraint.

1 Introduction

An important issue in GIR is the problem of *spatial querying* [2, 5, 3], intended as *supporting the distinct information needs of users that may access the same collection for different purposes*. To address it, GIRs must be developed to take user's preferences into account, to rank query results in terms of relevance [4].

In the *Geo-Finder* system [1], we devised a Geo-Retrieval model for flexible querying a GIR, such that: the user expresses the spatial condition based on the "close" soft constraint, adapting the *spatial scope* to the perceived meaning of spatial conditions; the user expresses preferences on how to combine the content conditions with the spatial conditions.

In the spatial condition, the user's context is modeled as user's perceived distance measure, that modifies the spatial scope of the query.

Two distinct semantics are provided for flexibly combining the content condition and the spatial condition: the asymmetric *and possibly* aggregation combines the mandatory content condition with the optional spatial condition; the compensative *average* aggregation linearly combines the two conditions. The relative weight between the conditions can be specified to achieve personalization.

2 The Geo-Retrieval model

In this paper, we present the Geo-Retrieval model devised in *Geo-Finder*. It is based on the concept of *Fuzzy Footprint*, that represents the degree with which a geographic reference is relevant for a document: for each indexed document, the *Geo-Indexer* [1] generates a set of fuzzy footprints.

A *fuzzy footprint* of a document d, denoted as $Foot(d)$, is a fuzzy set of geographic coordinates $\mathtt{gc}$= (lat,lon), where lat=latitude lon=longitude (expressed in degrees), with a membership degree $\mu_{Foot(d)}(\mathtt{gc}) \in [0,1]$ representing the significance by which the geographic location $\mathtt{gc}$ belongs to the geographic focus of document d:

$$Foot(d) = \{\langle\ \mathtt{gc}_1, \mu_{Foot(d)}(\mathtt{gc}_1)\rangle, \ldots, \langle \mathtt{gc}_n, \mu_{Foot(d)}(\mathtt{gc}_n)\rangle\}$$

where each $\mathtt{gc}_i = (lat_i,lon_i)$ and its membership degree $\mu_{Foot(d)}(\mathtt{gc}_i)$ are determined by the *Geo-Indexing* module [1].

A user query q consists of two conditions: a *content-based condition*, expressed by a list of content keywords, and a *spatial condition*, expressed by a list of geographic names. The spatial condition is interpreted as the requirement for documents with geographic reference "close" to the specified place names. These two conditions are evaluated by specific partial matching functions that compute two distinct scores in [0,1]: the *Retrieval Status Value* w.r.t. the content, denoted as $RSV_{content}(d)$, and the *Geographic Retrieval Value*, denoted as $GRV_{closeness}(d)$.
In *Geo-Finder*, $RSV_{content}(d)$ is a classical cosine similarity measure, computed by means of the *Lucene* library.

These two scores are finally combined to compute the *global Retrieval Status Value* w.r.t. the whole query q, indicated by $RSV_q(d)$, by applying a suitable aggregation function. We defined two aggregation functions, since we considered two distinct aggregation semantics, i.e., the *and possibly* asymmetric aggregation and the *average* compensative aggregation.

Evaluation of the spatial condition. Given the fuzzy footprint $Foot(q)$ of the geographic names in the query q, the fuzzy footprints of the documents d, $Foot(d)$, that are likely to satisfy the query are retrieved by accessing the *footprint spatial index*. The semantics of the spatial condition is that of evaluating a user's context dependent "closeness" of the documents' footprints $Foot(d)$ to the query footprint $Foot(q)$. This is done by a matching function `close` which models the concept of "close" as a user's context dependent soft constraint.

The matching function `close` computes a *Geographic Retrieval Value*, $GRV_{closeness}(d) \in [0,1]$, depending on the closeness of the document footprint to the query footprint as follows:

$$GRV_{closeness}(d) = \mu_{close}(Foot(d),Foot(q)) =$$
$$= max_{i \in Foot(d), j \in Foot(q)}\ qscope(dist(i,j) \times min(\mu_{Foot(d)}(i), \mu_{Foot(q)}(j)))$$

Where $\mu_{Foot(d)}(i)$ and $\mu_{Foot(q)}(j)$ are the membership degrees of the i-th and j-th fuzzy spatial references $\mathtt{gc}_i \in Foot(d)$ and $\mathtt{gc}_j \in Foot(q)$, i.e., the extent to which a spatial reference represents the geographic focus of the document and of the query, respectively.
The $dist(i,j)$ function is a great circle approximation of the actual distance between the two spherical coordinates $\mathtt{gc}_i$ and $\mathtt{gc}_j$.
The *qscope* function modifies the geographic distance so as to model the user perceived distance as follows:

$$qscope(x) = \begin{cases} \delta/(x+\delta) & if\ x \leq \delta + k \times MaxDist(Foot(d)) \quad with\ \delta \geq 0, k > 0 \\ 0 & otherwise \end{cases}$$

$MaxDist(X) = max_{i,j \in X}(dist(i,j))$ is the maximum geographic distance between any two geographic places i and j in the footprint X, and can be considered as the maximum dispersion of the fuzzy footprint X. It is zero in the case X contains just one single place. Thus $MaxDist(Foot(d))$ is the query dispersion. Its value depends on the number of geographic names specified in the query and on the maximum distance between their geographic coordinates.

The parameters δ and k permit to change the spatial scope of the query. The parameter δ is the *query range*, and is useful in the case of a query footprint consisting of a single geographic coordinate pair `gc` in order to retrieve also documents with footprint in the surrounding places. Distinct δ can adapt the evaluation of the spatial condition "close" to the user perception, thus, modeling strict or relaxed interpretations of the "closeness" surroundings of a point. The higher the δ, the greater is the surrounding.

The parameter k makes it possible to model a tolerance on the geographic distance between a document fuzzy footprint and the query footprint, so that one can consider close places within a distance of k times $MaxDist(Foot(d))$, i.e., k times the query maximum dispersion.

We consider four main query scopes that can be related to the user's context, and that are defined in the *Geo-Finder* system by the following default values of k and δ. (1) The *small scope* is defined with $k = 5$, $\delta = 3$ *km*; it is useful when $Foot(q)$ is a street address within a city or a small city and we are interested in its very near surroundings (in this case, $Foot(q)$ could vary approximately between 0 and about 10 *km*): with this setting, one can retrieve documents within a distance from the query of 3 *km* to about 50 *km*. (2) The *meso scope* is defined with $k = 4$, $\delta = 50$ *km*; in this case, $MaxDist(Foot(d))$ covers the area of either a region or a small nation like Belgium. (3) The *large scope* is defined with $k = 3$, $\delta = 1000$ *km*, in this case $MaxDist(Foot(d))$ covers the area of a medium nation such as France (in this case $Foot(q)$ could vary approximately between 0 and a few thousand kilometers). (4) The *full scope* is defined with $k = 3$, $\delta = 10000$ *km*; in thsi case, $MaxDist(Foot(d))$ covers the area of a big nation such as Russia or of a continent.

For example, if one specifies a spatial condition with the two geographic names *Bergamo*, *Como* (*Como* being at about 40 *km* from *Bergamo*), and the query scope is *meso* (i.e. $k = 4$ and $\delta = 50$ *km*) the documents with footprints at a maximum distance of 210 *km* from the query footprint are retrieved: for instance, both documents in *Milano* and *Lugano* are retrieved while a document with a footprint in *Rome* is not.

The Global RSV. Geo-Finder implements two distinct semantics to combine $RSV_{content}(d)$ and $GRV_{closeness}(d)$.

The asymmetric *and possibly* semantics is defined as follows:

$$RSV_q(d) = RSV_{content}(d) \text{ and possibly}^{\alpha} GRV_{closeness}(d) = \\ = RSV_{content}(d) \times max((1-\alpha), GRV_{closeness}(d))$$

Parameter α specifies the user's preference of the spatial condition w.r.t. the content condition. When $\alpha = 0$, it means that the spatial condition can be disregarded to rank the documents, and in this case the *global Retrieval Status Values* is determined solely based on the content relevance score $RSV_{content}(d)$.

When $\alpha = 1$, the two conditions are both mandatory: this means that the *Geographic Retrieval Value* $GRV_{closeness}(d)$ has the same relevance of the content *Retrieval Status Value* $RSV_{content}(d)$. In this case, the aggregation reduces to the product, i.e., the "fuzzy Anding" of the two relevance scores. Intermediate values of α in $(0, 1)$ demands for an asymmetric combination. The value $(1 - \alpha)$ guarantees a minimum satisfaction level for $GRV_{closenss}(d)$, so that the spatial condition becomes optional and the global $RSV_q(d)$ is not too much penalized in the case in which the spatial condition is not satisfied.

With the symmetric *Average* semantics, the *Global RSV* is defined as follows:

$$RSV_q(d) = RSV_{content}(d) \ average^{\alpha} \ GRV_{closeness}(d) =$$
$$= (1 - \alpha) \times \ RSV_{content}(d) + \alpha \times GRV_{closeness}(d)$$

When the preference degree $\alpha = 0$, the result is determined solely by the satisfaction of the content condition; conversely, when $\alpha = 1$, the global RSV is determined solely by the satisfaction of the spatial condition, and the content based condition is irrelevant. Intermediate values of α permit to vary the trade-off between the influences of the two conditions; in this case, the two conditions compensate each other, while with the *and possibly* semantics it is mandatory to satisfy the content condition to retrieve a document.

3 Conclusions

The Geo-Retrieval model described in this paper is implemented in the *Geo-Finder* system. In [1], we extensively presented its features. Furthermore, in [1], some evaluation results are also discussed showing the improvement of *Geo-Finder* ranking over *Google* ranking. The evaluations also showed that the precision of *Geo-Finder* improves when restricting the geographic domain of interest, thus outlining the positive role of modeling the user's context which determines the perceived distance when evaluating the spatial query condition.

References

1. G. Bordogna, G. Ghisalberti, and G. Psaila. Geographic information retrieval: Modeling uncertainty of user's context. *Fuzzy Sets and Systems*.
2. G. Cai. GeoVSM: An integrated retrieval model for geographic information. In *M.J. Egenhofer and D.M. Marks (Eds), GIScience 2002*, LNCS 2478, pages 65–79. 'Springer Verlag, 2002.
3. Z. Li, C. Wang, X. Xie, X. Wang, and W.Y. Ma. Indexing implicit locations for geographical information retrieval. In *n Proceedings of GIR-2006, Int. Conf. on Geographical Inf. Retrieval*, Seattle, USA, August 2006.
4. G. Mountrakis and A. Stefanidis. Moving towards personalized geospatial queries. *Journal of Geographic Information System*, 3:'334–344, 2011.
5. R.S. Purves, P. Clough, C.B. Jones, A. Arampatzis, B. Bucher, D. Finch, G. Fu, H. Joho, A.K. Syed, S. Vaid, and B. Yang. The design and implementation of SPIRIT: a spatially aware search engine for information retrieval on the internet. *International Journal of Geographical Information Science*, 21(7):'717–745, 2007.

Conversational Query Revision with a Finite User Profiles Model

Henry Blanco[1,2], Francesco Ricci[1], and Derek Bridge[3]

[1] Faculty of Computer Science
Free University of Bozen-Bolzano
Bolzano, Italy
fricci@unibz.it
[2] Center of Medical Biophysics
Universidad de Oriente
Santiago de Cuba, Cuba
[3] Department of Computer Science
University College Cork
Cork, Ireland

Abstract. Information Recommendation is a conversational approach aimed at suggesting to the user how to reformulate his queries to a product catalogue in order to find the products that maximize his utility. In previous work, it was shown that, by observing the queries selected by the user among those suggested, the system can make inferences on the true user utility function and eliminate from the set of suggested queries those retrieving products with an inferior utility (dominated queries). The computation of the dominated queries was based on the solution of several linear programming problems, which represented a major computational bottleneck for the efficiency of the proposed solution. In this paper we propose a new technique for the computation of the dominated queries. It relies on the assumption that the set of possible user utility functions is finite. We show that under this assumption the computation of the query suggestions is simplified and the number of query suggestions is strongly reduced.

Keywords: Recommender system, conversational system, user preference model.

1 Introduction

Recommender Systems (RS) are intelligent tools and applications designed to support users in finding information, products or services that suit their needs and preferences [8]. Recommender system technologies are rooted in Machine Learning and Information Retrieval [4]. The core computational problem of a RS is to predict the user's preferences, e.g. expressed as ratings for items, and recommend the items with maximal predicted preference [8]. Classical RS techniques, such as collaborative and content-based filtering, collect the user's preferences in the form of ratings for items to meet the goals mentioned before. Their

major limitation is that they present the recommendations in a single shot, and the user can either accept one of these recommendations or enter new preferences and restart the process. Conversely, Conversational Recommender Systems (CRS) [2, 5, 6] not only rank and suggest products to users, but also guide them during the human-computer interaction to finally select the products that they may like. This guidance process is composed of several actions that depend on the underlying conversational technology (e.g., critiquing [7, 6]).

In [1, 9] the authors first introduce and then extend a new conversational technique relying on the idea of "Information Recommendation". In this approach the user is supposed to query a product catalogue by issuing simple queries, such as "I want an hotel with AC and parking". The system, rather than recommending immediately the products that satisfy this query, assumes that the user may have also other needs and suggests some query revisions. These new queries, for instance, may add an additional feature to the query, e.g., the system may say: "are you interested also in a sauna?". Products with more features, if available, will surely increase the user utility. But not all features are equally important for the user. So the goal of the system is to make "informed" suggestions, i.e., to suggest features that are likely to increase more the user utility. In fact, the system, observing the user queries, can deduce that certain features are more important than others, i.e., can infer constraints on the definition of the user utility function, even without knowing it. Hence, using this knowledge, it can suggest that the user try a query from a well-selected and small set of candidate queries . A similar idea, i.e., using a utility function estimation to select the more user relevant critiques (new queries), is described in [10].

In [1][9] it is shown that this approach is effective and provides good query suggestions and final recommendations. It guides the user to the query that selects the products with maximal utility in a short number of query revision interactions. The quoted papers describe the details of the approach: the query language, the possible preference models of the user, the inferences made by the system on observing the user's query revisions, and the computation of the query suggestions for the user. Nevertheless some questions mostly related to the efficiency of the query suggestions computation and the size of the advice set are still open and require further investigation. In fact, the computational cost of query suggestion is playing a critical role in this approach. In [1] linear programming techniques were used for computing the query suggestions. Even if the computational complexity of that algorithm is polynomial, it must be invoked numerous times (to compare each pair of candidate queries), and in practice it takes too much time for a real online application. Moreover, the average size of the advice set, i.e., the queries suggested by the system to the user at each interaction step, remains large in many cases (more than 20). This is a critical issue for implementing a real application based on the proposed technique.

In this paper we refine the proposed model by making the assumption that the user utility function is drawn from a set of finite possibilities. This set of "user profiles" represents the possible "different" users that the system may interact with. We will show that this assumption has a strong effect: it simplifies

the search process for the query suggestions and reduces the average number of query suggestions made at each interaction step. This finite model assumption is realistic, as users tend to cluster in groups with similar preferences. Moreover, considering an increasingly large number of user profiles one can approximate all the possible ones.

2 Query Language

In our model a product p is represented by an n-dimensional Boolean feature vector $p = (p_1, \ldots, p_n)$. $p_i = 1$ means that the i-th feature (e.g., Air Conditioning) is present in the product, whereas $p_i = 0$ means that p does not have feature i. A catalogue is a set of products $\{p^{(1)}, \ldots, p^{(k)}\}$. The Boolean features could be keywords or tags found in the product description, and searching for products with these features can be viewed as kind of facet search [3].

Queries are represented similarly as Boolean vectors: $q = (q_1, \ldots, q_n)$. $q_i = 1$ means that the user is interested in products that have the i-th feature. On the other hand $q_i = 0$ does not mean that the user is not interested in products with that feature, but simply that he has not yet declared his interest on it. A query is said to be *satisfiable* if there exists a product in the catalogue such that all the features expressed in the query as desired ($q_i = 1$) are present in that product. For example if the product $p = (1, 1, 0, 1, 0)$ is present in the catalog then query $q = (0, 1, 0, 1, 0)$ is satisfiable.

We are considering a scenario where the user may be interested in refining an initial query. Moreover, we assume that the user is not likely to radically modify this query. This may also be a constraint imposed by the GUI of the query system, where the user can be offered with only a small number of easily understood editing operations. In the following we list the query editing operations that we assume the user can make when revising the current query:

- $add(q, i)$, where $i \in idx0(q)$
- $trade(q, i, j, k)$, where $i \in idx1(q)$ and $j, k \in idx0(q)$

where $idx0(q)$ and $idx1(q)$ are the set of indexes with value 0 and 1 in q respectively. The first operation generates a new query by requesting one additional feature. For example, $(1, 1, 0, 0, 1) = add((1, 1, 0, 0, 0), 5)$ is extending a query where only the first two features were requested by adding also the fifth feature to the set of requested ones. The second operation generates a new query by discarding a feature, the i-th, in favor of two new ones, the j-th and k-th features. For example, $(0, 1, 0, 1, 1) = trade((1, 1, 0, 0, 0), 1, 4, 5)$

Using the above-mentioned operators the system can generate a set of next queries and ask the user to select the preferred one. In our approach, the goal of the system is not to suggest all these possible next queries, as a standard "query by example" interface may implement, but rather only queries that could retrieve products with the largest utility. Hence, first of all, the unsatisfiable queries must not be suggested. This can be easily implemented with standard query processing techniques. But, as it will be shown later, also other types of queries

can be discarded: those that can be proved to retrieve products with a smaller utility than those retrieved by another query in the suggestion list (dominated queries).

3 User Utility Function

User preferences for products are represented here as a vector of weights:

$$w = (w_1, \ldots, w_n), 0 \leq w_i \leq 1 \tag{1}$$

w_i is the importance that a particular user, one having that set of preference weights, assigns to the i-th feature of a product. So if $w_i = 0$, then the user has no desire for the i-th feature. If $w_i > w_j$, then the i-th feature is preferred to the j-th one. If $w_i \geq w_j$ then the i-th feature is at least as desired as the j-th one. If $w_i = w_j$, $i \neq j$ then the user is indifferent between these two features. The user utility for a particular product $p = (p_1, \ldots, p_n)$ is given by the following:

$$Utility_w(p) = \sum_{i=1}^{n} w_i \times p_i \tag{2}$$

A product p with a higher utility than another product p' is always assumed to be preferred by the user, i.e., we assume that users are rational. We also define the potential utility of a query $q = (q_1, \ldots, q_n)$ for the user as: $Utility_w(q) = \sum_{i=1}^{n} w_i \times q_i$. We call this utility "potential" if we do not know wether a product with the features specified in the query does exist, i.e., if the query is satisfiable. In case such a product exists, this potential utility is also a true utility.

A user accessing the system may have any of the possible utility functions that can be defined by varying the feature weights w_i. So, in principle, the set of all possible utility functions is infinite. But observing the queries selected by the user among those that he can make (i.e., those suggested by the system), the system can infer constraints on the definition of his utility function. Generally speaking, features present in the selected query can be considered as more desired by the user than features that are present in the alternative queries. The constraints deduced by the system on the user utility function $w = (w_1, \ldots, w_n)$ are illustrated below.

Initial query. If the current query q is the initial query, then the advisor may infer that $w_i \geq w_j$, $\forall i \in idx1(q)$ and $\forall j \in idx0(q)$, unless q, with the i-th feature set to 0 and the j-th feature set to 1, is unsatisfiable. This means that if the user issued a query that requests the presence of a feature then the potential utility of this query is assumed to be larger than or equal to that of another query where this feature is not requested. But only if this "alternative query" is satisfiable.

Adding a feature. If the current query q' results from an $add()$ operation on the previous query, that is, $q' = add(q, i)$, then the advisor infers $w_i \geq w_j$, $\forall j \in idx0(q)$, $i \neq j$, unless $add(q, j)$ results in an unsatisfiable query. The rationale of this deduction is similar to the previous one. We assume that the user has

extended the query by selecting a new query that includes an additional feature that brings a larger increase of his utility, compared to the other possible features that he may have included.

Trading one feature for two. If the current query results from a trade operation on the previous query, i.e., $q' = trade(q, i, j, k)$, the advisor may infer:

1. $w_j + w_k \geq w_i$,
2. $w_j + w_k \geq w_{j'} + w_{k'}, \forall j', k' \in idx0(q), \{j, k\} \neq \{j', k'\}$ unless $trade(q, i, j', k')$ is unsatisfiable.

The first constraint says that the current query does not have a utility inferior to the previous one. While the second constraint says that the selected trade operation must obtain a utility that is not inferior to that of alternative trade operations that the user may have applied (and are satisfiable).

We note that unsatisfiable queries are never suggested, and therefore we never deduce that a query has a potential utility larger than that of a failing query. In the previous work [1], we called this "play safe" because we considered that the user might know that a query will fail and therefore he does not try it, hence we cannot assume that the potential utility of the query that was actually tried is larger than that of a query that the user did not try because he knew it would fail. In the current work we generalize and rephrase it by saying that the system can deduce only that the potential utility of the query that is tried is greater than or equal to the (potential) utility of the other queries that were suggested, or equivalently that the user could have tried (either because the system suggested them or because the user knows they are satisfiable).

4 Advisor

The advisor is the intelligent entity in charge of observing the interaction process, the user movements (queries issued), and making inferences on the user preferences. As mentioned before, the user preferences are not known at the beginning of the interaction between the user and the advisor. The advisor, after the user's first query, will generate a set of next candidates queries and will suggest only those with a utility that cannot be proved to be inferior to one of the other queries (undominated queries).

At each user-system interaction step, the advisor accumulates some constraints on the user utility function (see Section 3). We denote this set of constraints by Φ. Moreover, given a set of next possible queries $C = \{q^{(1)}, \ldots, q^{(k)}\}$, i.e., those that can be generated by applying the operations described in Section 2, and that are satisfiable, the advisor needs to understand which queries are worth suggesting to the user. These are the queries having a utility not inferior to the utility of another query that may also be suggested. These queries are obtained by removing from C all the dominated queries.

A query $q \in C$ is *dominated* if there exists another query $q' \in C$ such that for all the possible weight vectors that are compatible with the set of constraints Φ this relation holds: $Utility_w(q') > Utility_w(q)$. A weight vector w is said to be

Table 1. Query utilities for the profiles $w^{(1)}$ and $w^{(3)}$.

	$\mathbf{q^{(1)}}$	$\mathbf{q^{(2)}}$	$\mathbf{q^{(3)}}$	$\mathbf{q^{(4)}}$
$\mathbf{w^{(1)}}$	0.75	0.9	0.65	0.7
$\mathbf{w^{(3)}}$	0.9	0.65	0.7	0.75

compatible with the set of constraints in Φ if and only if all the constraints in Φ are satisfied when the variables $w_1, \ldots, w_n$ take the values specified in w.

Removing the dominated queries is meaningful because their utility is lower than the utility of another query (that is suggested) for all the possible user utility functions that are compatible with the preferences induced by observing the user behavior. In this paper we solve this problem under the assumption that the user's true utility function is defined by one (unknown) vector among a finite set of weights vectors considered by the system. We call this finite set of all the possible utility function or "user profiles" $P = \{w^{(1)}, \ldots, w^{(m)}\}$. We will consider in the experiments m ranging from some dozens to hundreds.

With this assumption, having the set Φ we can prune from the set P the "incompatible profiles", i.e., those not satisfying the constraints Φ. Then, the computation of the undominated queries proceeds as follow. Let's assume that the set of user profiles compatible with the accumulated constraints is $P' = \{w^{(1)}, \ldots, w^{(t)}\} \subset P$ and $C = \{q^{(1)}, \ldots, q^{(k)}\}$ is the set of next possible queries, i.e., queries that are satisfiable and are generated by the considered operators starting from the last issued query of the user. The final set of queries that are recommended are computed using a linear time procedure in the number of queries in C and utility functions in P', as follows:

1. A query $q \in C$, is labelled as dominated if and only if we can find another query $q' \in C$, $q' \neq q$, such that $\forall w \in P'$, $Utility_w(q') > Utility_w(q)$, i.e., $\sum_{i=1}^{n} w_i \times q'_i > \sum_{i=1}^{n} w_i \times q_i$.
2. Build the Advice set - undominated queries - by removing from C the dominated queries.

Example. Assume that $\Phi = \{w_1 \geq w_3, w_2 + w_3 \geq w_4\}$, $P' = \{w^{(1)}, w^{(2)}, w^{(3)}\}$ and $C = \{q^{(1)}, q^{(2)}, q^{(3)}, q^{(4)}\}$, $w^{(1)} = (0.35, 0.1, 0.25, 0.3)$, $w^{(2)} = (0.1, 0.35, 0.3, 0.25)$, $w^{(3)} = (0.3, 0.35, 0.1, 0.25)$, $q^{(1)} = (1, 1, 0, 1)$, $q^{(2)} = (1, 0, 1, 1)$, $q^{(3)} = (0, 1, 1, 1)$, $q^{(4)} = (1, 1, 1, 0)$. In this example only the profiles $w^{(1)}$ and $w^{(3)}$ satisfy the constraints in Φ, so $w^{(2)}$ is an "incompatible profile", and must be pruned from P'. Table 1 shows the query utilities. $q^{(1)}$ has a higher utility than $q^{(3)}$ and $q^{(4)}$ for every profile in P', thus $q^{(3)}$ and $q^{(4)}$ are dominated by $q^{(1)}$. These dominated queries are removed from the set C. Notice that the remaining queries $q^{(1)}$ and $q^{(2)}$ do not dominate each other, thus they represent meaningful advice that the advisor can provide to the user.

Finally, the algorithm for query suggestions using a finite set of user profiles is described as follows:

1. $\Phi = \emptyset$, P = all possible profiles, AdviceSet = all possible queries
2. **Do**
3. Present AdviceSet to the user;
4. currentQuery = query selected by the user in AdviceSet;
5. Infer constraints analyzing the currentQuery and add them to Φ;
6. Remove incompatible profiles from P;
7. Compute candidate queries;
8. Remove dominated queries from candidate ones and generate AdviceSet;
9. **while** ((AdviceSet $\neq$ null) and (user wants advice))

The advisor presents to the user a possible set of queries. At the beginning these are all the possible ones, i.e., the user is free to enter the first query. Then the advisor infers the constraints Φ according to the rules mentioned in section 3. The advisor then removes the user profiles that do not satisfy these constraints. Afterwards the set of candidate queries are generated from the current query, applying the operators mentioned in Section 2 and those that are not satisfiable are removed. Finally, the advisor identifies the AdviceSet by removing the dominated queries and suggests the remaining ones to the user as potential new moves. If the user selects one from this advice and the AdviceSet is not empty then the selected query becomes the current query and the process is repeated. If the user does not want further advice then the system will suggest the products that satisfy the last query selected by the user.

5 Experiments

We performed some experiments in order to compare the performance of the proposed approach with the results obtained in [1]. We simulated several interactions between a user and the advisor. We varied the following parameters in the simulations: the product database and the number and format of the user profiles. Three different product databases were used, each one describing real hotels by their amenities expressed as Boolean features. Details of the product databases are given in the Table 2; here an hotel may have the same product description in terms of features as another, which is why the number of distinct products is smaller than the number of hotels.

We considered two kinds of user profiles as typical models of user preferences: "random-shape user profiles" and "exponential-shape user profiles". A "user's profile shape" refers to the distribution of the weights of the features in a user profile. Random-shape user profiles are created by first generating one initial user profile (weights vector) sampling the weights from a uniform distribution in [0,1]. Then the other profiles, in the same set P, are created by a random permutation of the feature weights of the initial user profile. Note that if the weights are sorted into decreasing order, the resulting sequence will decrease near linearly. This is because there is no special 'preference' for any number when you randomly select them. Conversely, the set of exponential-shape user profiles is created by generating first one initial user profile with an exponentially decreasing importance for the weights: $e^{-\alpha i}$, with a selected $\alpha \in [1, 4]$ and $i =$

Table 2. Product databases

Name	Features	Hotels	Products
Marriot-NY	9	81	36
Cork	10	21	15
Trentino-10	10	4056	133

$1, \ldots, n$. The other user profiles are again obtained with random permutations of the initial user profile. Here we wanted to simulate users with a few important features and many less important ones. For each experiment, we generated three sets of user profiles P: small (24 profiles), medium (120 profiles) and large (720 profiles). We wanted to observe the effect of the assumed variability of the user profiles on the user-advisor interaction length and the size of the advice set.

We assumed that the user is "Optimizing" [1], that is, one who confines his queries to the advice set provided by the advisor and he will always try the query with the highest utility in the advice set. The simulated interaction between a virtual user and the advisor is done considering the algorithm described in the previous section. One element of the set of predefined user profiles is randomly selected and considered as the user's true profile (user's utility). This is not revealed to the advisor, which interacts with the simulated user using the proposed methodology. The advisor deductions about the user's true utility function are based only on the observation of the user queries submitted at each interaction step. The initial query submitted by the simulated user is created in accordance to his true utility function; thus, the initial query includes up to the k most important features for the user.

In total, 18 experiments were performed corresponding to the combination of the variables mentioned before (product database, user profile shape and number of user profiles). In every experiment we ran 50 dialogues between a simulated user and the advisor. The observed measures were: the average number of queries issued per dialogue, the average size of the advice set and the average utility shortfall. The utility shortfall is the difference between the utility of the best query (selecting the product with the highest utility for the user) and the last query suggested by the system to the user. In this way we could measure if the system suggestions are close to the best query according to the user's true utility function.

Table 3 shows the values of the observed measures. We can observe that the average number of queries issued by the virtual user (interaction length) ranges between 3 and 7 almost independently from the "User Profile shape" and "User Profile set size". The interaction length seems to be related to the number of product features and the available products in the data set. The higher the number of product's features the longer will be the interaction. This happens because the user at each query editing step adds one feature to the previous query. In fact, the query suggestions are generated by the *add*() and *trade*() operations that extend the previous query by setting one additional feature to

Table 3. Averaged values of the observed measures for 50 runs in the 18 experiments performed.

Product D.B.	User Prof. shape	User Prof. set size	Queries issued	Queries in Adv. set	Utility Shortfall
Cork	Random	24	5.58	1.21	0.00281
		120	5.21	2.83	0.00491
		720	5.64	4.71	0.00622
	Exponential	24	5.66	1.28	0.0082
		120	5.52	3.05	0.0062
		720	5.31	3.37	0.0051
Marriott	Random	24	4	1.59	0
		120	4	3.41	0.00031
		720	4	4.33	0.00083
	Exponential	24	3.67	1.67	0.00636
		120	4	3.01	0.00097
		720	4	5.73	0
Trentino	Random	24	6.62	1.08	0.00315
		120	6.58	2.06	0.00319
		720	6.32	2.93	0.00757
	Exponential	24	6.38	1.14	0.00019
		120	6.18	1.72	0.00197
		720	6.48	2.89	0.00833

1. Hence, assuming that the best query has a certain number of features set to 1, then the user needs to pass through that number of steps (minus the number of features set to 1 in the initial query) in order to reach it, or to reach another query that does not provide the maximal utility but still cannot be further extended without reaching a failing query. Another factor to take into account is the number of products in the database. The smaller the number of products is, the more likely the process is to stop, because the current query cannot be further extended without building a failing query. The most important aspect of these values is that the interaction length is typically low and quite reasonable for real online applications.

The "Average size of the advice set" is sub-linearly correlated to the profile set size, that is, to the number of predefined user profiles. The higher the number of predefined user profiles, the (slightly) higher is the number of query suggestions in the advice set on average. In fact, if there are more user profiles, the more difficult it is to find dominated queries, thus the set of undominated queries (the advice set) is more likely to be larger. In general the average advice set size ranges between 1 and 6. This number of query suggestions represents an acceptable value for real applications. The "Average size of the advice set" doesn't seem to be related to the variables "User Profile shape" and "Product database". In

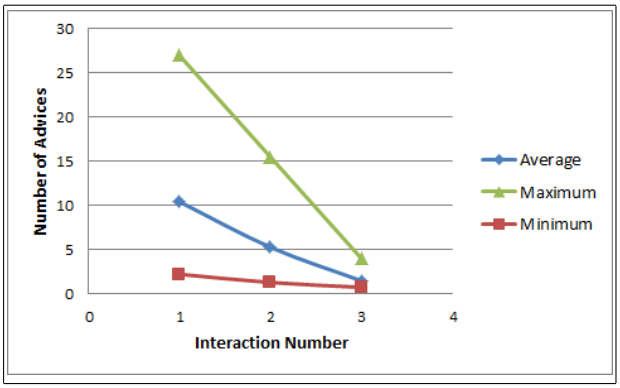


Fig. 1. Size of the Advice Set at different interaction steps.

general the "User profile shape" doesn't seem to influence either the "Interaction length" or the "Average size of the advice set".

The utility shortfall is very close to 0 on average. This cannot be 0 because the query suggestions are searched in a greedy way (always expanding the previous query), hence the advisor can fall into local maxima paths while searching for the best query suggestion [1]. Thus we cannot assure that the Advice Set will always contains the query with the largest utility that can be obtained by using the current query editing operations. Hence, limiting the query editing to the $add(\ldots)$ and $trade(\ldots)$ operators does not assure the user to reach the best query. Nevertheless at the end of the process the final query is very close to the best attainable given the user preferences.

Figure 1 shows the evolution of the Advice Set size (averaged over 50 dialogues) in the experiment that produces the highest number of average advices per suggestion (5.73 queries in table 3). That experiment corresponds to: Product Database = Marriott NY, Profile Shape = Exponential, and User Profile set size = 720. The curve labeled as "average" shows the average number of advices given to the user at the first three interaction steps. At the first step, the number of queries suggested is on average $10.4 \pm 8.2 (avg. \pm stdv.)$; at the next interaction step, it is 5.3 ± 3.9; and finally the system suggests only 1 ± 0.7 queries (the best). The curves labeled as "Maximum" and "Minimum" correspond to the maximum and minimum number of queries suggested at each interaction step to the user. In general we can see that the number of advices falls quickly in a short number of interactions. Still, it is clear that there are certain dialogues with a rather large number of advices, and this is an issue to consider in the application of this technique.

We now compare our results with those presented in [1]. Table 4 shows the values of the variables "Average number of queries issued per Dialogue", "Average size of the Advice Set", "Average Utility shortfall" obtained in the previous work, where an infinite number of profiles was considered and the query dom-

Table 4. Comparison between the current (finite model) and previous work (infinite model) on the observed measures.

Database	Averaged measures	Infinite model	Finite model
Marriott-NY	Queries issued	4.67	3.58
	Numb. Advices	45.96	4.33
	Utility shortfall	0	0.0008
Cork	Queries issued	6.09	6.32
	Numb. Advices	69.88	2.93
	Utility shortfall	0	0.0075
Trentino	Queries issued	5.55	5.64
	Numb. Advices	59.02	4.71
	Utility shortfall	0	0.0062

inance relation was computed using linear programming techniques. It is clear that the average number of queries per dialogue is low in both approaches and very similar. This is due to the fact that the actual query editing operations are the same in the two approaches, and the dialogues converge to optimal queries with similar operations. The utility shortfall in the current approach is a bit larger than that measured previously. This is what one has to pay for the limiting assumption that the number of possible user utility functions (profiles) is finite. The major beneficial effect of the proposed approach is the significant reduction in the number of queries suggested by the advisor to the user by more than 10 times. This makes it much more suitable in real applications. Obviously this is again related to the assumption that the variability of the user utility functions is assumed to be smaller. We believe that in real scenarios approximating the set of all possible utility functions with a smaller, finite set, is a reasonable assumption and the small cost paid in terms of increased utility shortfall is compensated by the strong reduction in the size of the advice set, making it feasible for the user to browse the advice set and pick up his best query.

6 Conclusions and Future Work

In this paper we have described and analyzed the performance of a new type of conversational recommender system that suggests query revisions to a user searching for products in a catalogue. The products are described by Boolean features. They can be for instance tags or keywords found in the product descriptions. In this paper we assume that the user utility function is one among a finite set of possible functions that are known to the system, but the system does not know which is the true utility function of the user.

The results of our experiments showed that this assumption has a strong effect on the process of finding the best query suggestions that guide the user to the products that maximize his utility. In particular the number of user-advisor interaction steps (number of queries issued by the user) and the utility

shortfall are low (as in our previous work where the user profiles were not limited to be finite). But, differently from the previous case, we have now observed a significant reduction in the number of advices provided at each user-advisor interaction step. We have also showed that having a good number of predefined user profiles is an important ingredient for improving the system performance and producing an effective support.

In future work we will consider the case when the true utility function of the user is not one of those assumed by the system. This is the true general situation when a totally unknown user is approaching the system and the system has no knowledge about his preferences. In particular, we will measure how this impacts on the utility shortfall. Additionally we will implement this approach on a real online and mobile application, which will undoubtedly help to give a better understanding of user behavior and the true effectiveness of the proposed approach.

References

1. D. Bridge and F. Ricci. Supporting product selection with query editing recommendations. In *RecSys '07: Proceedings of the 2007 ACM conference on Recommender systems*, pages 65–72, New York, NY, USA, 2007. ACM Press.
2. M. H. Göker and C. A. Thomson. Personalized conversational case-based recommendation. In *Advances in case-based reasoning: 5th European workshop, EWCBR-2000, Trento, Italy, September 6–9, 2000: proceedings*, pages 99–111. Springer, 2000.
3. M. A. Hearst. *Search User Interfaces*. Cambridge University Press, 2009.
4. P. Lops, M. de Gemmis, and G. Semeraro. Content-based recommender systems: State of the art and trends. In F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, editors, *Recommender Systems Handbook*, pages 73–105. Springer Verlag, 2011.
5. T. Mahmood and F. Ricci. Learning and adaptivity in interactive recommender systems. In *ICEC '07: Proceedings of the ninth international conference on Electronic commerce*, pages 75–84, New York, NY, USA, 2007. ACM.
6. L. McGinty and J. Reilly. On the evolution of critiquing recommenders. In F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, editors, *Recommender Systems Handbook*, pages 419–453. Springer Verlag, 2011.
7. Q. N. Nguyen and F. Ricci. User preferences initialization and integration in critique-based mobile recommender systems. In *Proceedings of the 5th International Workshop on Artificial Intelligence in Mobile Systems, AIMS'04*, pages 71–78, Nottingham, UK, 2004.
8. F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, editors. *Recommender Systems Handbook*. Springer, 2011.
9. W. Trabelsi, N. Wilson, D. Bridge, and F. Ricci. Comparing approaches to preference dominance for conversational recommender systems. In E. Gregoire, editor, *Procs. of the 22nd International Conference on Tools with Artificial Intelligence*, pages 113–118, 2010.
10. J. Zhang and P. Pu. A comparative study of compound critique generation in conversational recommender systems. In *Proceedings of the 4th International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, AH 2006*, pages 234–243, 2006.

Uncertain Graphs meet Collaborative Filtering

Claudio Taranto, Nicola Di Mauro, and Floriana Esposito

Department of Computer Science, University of Bari "Aldo Moro"
via E. Orabona, 4 - 70125, Bari, Italy
{claudio.taranto,ndm,esposito}@di.uniba.it

Abstract. Collaborative filtering (CF) aims at predicting the user interest for a given item. In CF systems a set of users ratings is used to predict the rating of a given user on a given item using the ratings of a set of users who have already rated the item and whose preferences are similar to those of the user. In this paper we propose to use a framework based on *uncertain graphs* in order to deal with collaborative filtering problems. In this framework relationships among users and items and their corresponding likelihood will be encoded in a uncertain graph that can then be used to infer the probability of existence of a link between an user and an item involved in the graph. In order to solve CF tasks the framework uses an approximate inference method adopting a constrained simple path query language. The aim of the paper is to verify whether uncertain graphs are a valuable tool for CF, by solving classical, complex and structured problems. The performance of the proposed approach is reported when applied to a real-world domain.

1 Introduction

The inherent uncertainty and complexity present in some real world domains has led to the emerging of many probabilistic frameworks, such as probabilistic graphical models [14] and statistical relational learning [6], able to deal with uncertain and structured domains. Learning and reasoning on *uncertain graphs*[1] has become an increasingly important research topic [19, 29, 9, 11]. In this model, each edge is associated with a probability representing the likelihood of its existence in the graph, and the edges existence is assumed to be mutually independent.

Collaborative filtering (CF) aims at predicting the user interest for a given item based on a collection of user profiles. Collaborative filtering is an approach adopted in recommender systems that attracted much of attention in recent years. In CF systems a set of users ratings is used to predict the rating of a given user u on a given item i using the ratings of a set of users who have already rated i and whose preferences are similar to the ones of u.

CF systems need to compare items against users and this task may be solved with a *memory based* approach that may be divided into *user-based* or *item-based* approaches. A typical example of memory based approaches are *neighborhood*

[1] Uncertain graphs are also referred to *probabilistic graphs* as in [29, 9].

based CF methods centered on computing the relationships between items or between users. Given an unknown rating to be estimated, memory-based CF firstly computes similarities between the given user and other users (*user-based* approach), or between the given item and other items (*item-based* approach). Then, the unknown rating is predicted by averaging the known ratings by similar users or by similar items [4, 15].

In this paper we propose to use uncertain graphs to deal with collaborative filtering problems. In particular, relationships among users and items and their corresponding likelihood will be encoded in a uncertain graph that can then be used to infer the probability of existence of a link between an user and an item involved in the graph.

The main questions that we want to answer in this paper are the following:

- **Q1**: are uncertain graphs a valuable tool for collaborative filtering?
- **Q2**: can uncertain graphs solve classical CF user-based and item-bases tasks?
- **Q3**: can uncertain graphs unify user-based and item-based CF approaches?

2 Uncertain graphs

Let $G = (V, E)$, be a graph where V is a collection of nodes and $E \in V \times V$ is the set of edges, or relationships, between the nodes.

Definition 1 (Uncertain graph). *An* uncertain graph *is a system* $G = (V, E, \Sigma, l_V, l_E, P)$*, where* (V, E) *is an undirected graph,* V *is the set of nodes,* E *is the set of edges,* Σ *is a set of labels,* $l_V : V \to \Sigma$ *is a function assigning labels to nodes,* $l_E : E \to \Sigma$ *is a function assigning labels to the edges, and* $P : E \to [0, 1]$ *is a function assigning* existence probability *values to the edges.*

The existence probability $P(e)$ of an edge $e = (u, v) \in E$ is the probability that edge between u and v can exist in the graph. A particular case of uncertain graph is the *certain graph* when the existence probability value on all edges is 1. In this paper we use the possible world semantics. In particular, we can imagine an uncertain graph G as a sampler of worlds, where each world is an instance of G. A certain graph G' is sampled from G according to P, denoted as $G' \sqsubseteq G$, when each edge $e \in E$ is selected to be an edge of G' with probability $P(e)$. Edges labeled with probabilities are treated as mutually independent random variables indicating whether or not the corresponding edge belongs to a certain graph. Assuming independence among edges, the probability distribution over certain graphs $G' = (V, E') \sqsubseteq G = (V, E)$ is given by

$$P(G'|G) = \prod_{e \in E'} P(e) \prod_{e \in E \setminus E'} (1 - P(e)). \tag{1}$$

Definition 2 (Simple path). *Given an uncertain graph* G*, a* simple path *of a length* k *from* u *to* v *in* G *is a sequence of edges* $p_{u,v} = \langle e_1, e_2, \ldots e_k \rangle$*, such that* $e_1 = (u, v_1)$*,* $e_k = (v_{k_1}, v)$*, and* $e_i = (v_{i-1}, v_i)$ *for* $1 < i < k$*, and all nodes in the path are distinct.*

Given G an uncertain graph, and $p_{s,t}$ a path in G from node s to node t, $l(p_{s,t}) = l(e_1)l(e_2)\cdots l(e_k)$ denotes the concatenation of the labels of all edges in $p_{s,t}$. Given a *context free grammar* (CFG) $\mathcal{C}$ a string of terminals s is derivable from $\mathcal{C}$ iff $s \in L(\mathcal{C})$, where $L(\mathcal{C})$ is the language generated from $\mathcal{C}$.

Definition 3 (Language constrained simple path). *Given an uncertain graph G and a context free grammar $\mathcal{C}$, a* language constrained simple path *is a simple path p such that $l(p) \in L(\mathcal{C})$.*

Given an uncertain graph G a main task corresponds to compute the probability that there exists a path between two nodes u and v, that is, querying for the probability that a randomly sampled certain graph contains a path between u and v. More formally, the *existence probability* $P_e(q|G)$ of a path q in an uncertain graph G corresponds to the marginal $P(G'|G)$ with respect to q:

$$P_e(q|G) = \sum_{G' \sqsubseteq G} P(q|G') \cdot P(G'|G) \tag{2}$$

where $P(q|G') = 1$ if there exits the path q in G', and $P(q|G') = 0$ otherwise. In other words, the existence probability of path q is the probability that the path q exists in a randomly sampled certain graph.

Definition 4 (Language constrained simple path probability). *Given an uncertain graph G and a context free grammar $\mathcal{C}$, the* language constrained simple path probability *of $L(\mathcal{C})$ is*

$$P(L(\mathcal{C})|G) = \sum_{G' \sqsubseteq G} P(q|G', L(\mathcal{C})) \cdot P(G'|G) \tag{3}$$

where$P(q|G', L(\mathcal{C}) = 1$ if there exists a path q in G' such that $l(q) \in L(\mathcal{C})$, and $P(q|G', L(\mathcal{C})) = 0$ otherwise.

In particular, the previous definition give us the possibility to compute the probability of a set of simple path queries fulfilling the structure imposed by a context free grammar. In this way we are interested in certain graphs that contain at least one path belonging to the language corresponding to the given grammar.

2.1 Inference

Computing the existence probability directly using (2) or (3) is intensive and intractable for large graphs since the number of certain graphs to be checked is exponential in the number of probabilistic edges. It involves computing the existence of the path in every certain graph and accumulating their probability. A natural way to overcome the intractability of computing the existence probability of a path is to approximate it using a Monte Carlo sampling approach [12]: 1) we sample n possible certain graphs, $G_1, G_2, \ldots G_n$ from G by sampling edges uniformly at random according to their edge probabilities; and 2) we check if the

path exists in each sampled graph G_i. This process provides the basic sampling estimator

$$\widehat{P_e}(q|G) \approx P_e(q|G) = \frac{\sum_{i=1}^{n} P(q|G_i)}{n} \tag{4}$$

Note that is not necessary to sample all edges to check whether the graph contains the path. For instance, assuming to use an iterative depth first search procedure to check the path existence. When a node is just visited, we will sample all its adjacent edges and pushing them into the stack used by the iterative procedure. We will stop the procedure either when the target node is reached or when the stack is empty (non existence).

3 Uncertain graphs for collaborative filtering

The most common approach to CF is based on neighborhood models. User-oriented methods estimate unknown ratings based on recorded ratings of similar users, while in item-oriented approaches ratings are estimated using known ratings made by the same user on similar items.

Let U be a set of n users and I a set of m items. A rating r_{ui} indicates the preference by user u of item i, where high values mean stronger preference. Let S_u be the set of items rated from user u. For user-based approaches, the prediction of an unobserved rating $\widehat{r_{ui}}$ is computed as follows

$$\widehat{r_{ui}} = \overline{r_u} + \frac{\sum_{v \in U | i \in S_u} s_{uv} \cdot (r_{vi} - \overline{r_v})}{\sum_{v \in U | i \in S_u} |s_{uv}|} \tag{5}$$

where $\overline{r_u}$ represents the mean rating of user u, and s_{uv} stands for the similarity between users u and v, computed, for instance, using the Pearson correlation:

$$s_{uv} = \frac{\sum_{a \in S_u \cap S_v} (r_{ua} - \overline{r_u}) \cdot (r_{va} - \overline{r_v})}{\sqrt{\sum_{a \in S_u \cap S_v} (r_{ua} - \overline{r_u})^2 \sum_{a \in S_u \cap S_v} (r_{va} - \overline{r_v})^2}} \tag{6}$$

On the other side, item-based approaches predict the rating of a given item using the ratings of the user on the items considered as similar to the target item. Given a similarity measure, such as the Pearson correlation, the rating $\widehat{r_{ui}}$ is estimated as:

$$\widehat{r_{ui}} = \frac{\sum_{j \in S_u | j \neq i} s_{ij} \cdot r_{uj}}{\sum_{j \in S_u | j \neq i} |s_{ij}|} \tag{7}$$

These neighbourhood approaches see each user connected to other users or consider each item related to other items as in a network structure. In particular they rely on the direct connections among the entities involved in the domain. However, as recently proved, techniques able to consider complex relationships among the entities, leveraging the information already present in the network, involves an improvement in the processes of querying and mining [24, 21]. In [24] the authors improved the accuracy of a similarity measures between

two annotated nodes in a graph by using link information. They showed that the similarity between nodes annotations may be improved using also the network context. Another approach [20] to enriched a graph representation is the addition of semantic information improving link prediction results in network datasets. In particular, a supervised learning method for building link predictors from structural attributes of the underlying network using some semantic attributes of the nodes has been adopted.

The approach used in this paper is to represent a dataset consisting of user ratings, $\mathcal{K} = \{(u, i, r_{ui}) | r_{ui} \text{ is known}\}$, with an uncertain graph and then performing inference on this graph to solve classical collaborative filtering tasks. Hence the question to be solved is how to build the uncertain graph from the flat rating representation $\mathcal{K}$. The formal characterization we have provided about uncertain graphs gives us the possibility to represent heterogeneous objects and connections.

3.1 Uncertain graph construction

Given the set of ratings $\mathcal{K} = \{(u, i, r_{ui}) | r_{ui} \text{ is known}\}$, we add a node with label `user` for each user in $\mathcal{K}$, and a node with label `item` for each item in $\mathcal{K}$. The next step is to add the edges among the nodes. Each edge is characterized by a label and a probability value, which should indicate the degree of similarity between the two nodes. Two kind of connections between nodes are added. For each user u, we added an edge, labeled as `simU`, between u and the k most similar users to u. The similarity between two users u and v is computed adopting a weighted Pearson correlation between the items rated by both u and v.

In particular, the probability of the edge `simU` connecting two users u and v is computed as:

$$P(\texttt{simU}(u, v)) = s_{uv} \cdot w_u(u, v),$$

where s_{uv} is the Pearson correlation between the vectors of ratings corresponding to the set of items rated by both user u and user v, and

$$w_u(u, v) = \frac{|S_u \cap S_v|}{|S_u \cup S_v|},$$

where S_u is the set of items rated from user u.

For each item i, we added an edge, with label `simI`, between i and the most k similar items to i. In particular, the probability of the edge `simI` connecting the item i to the item j has been computed as:

$$P(\texttt{simI}(i, j)) = s_{ij} \cdot w_i(i, j),$$

where s_{ij} is the Pearson correlation between the vectors corresponding to the histogram of the set of ratings for the item i and the item j, and

$$w_i(i, j) = \frac{|\overline{S}_i \cap \overline{S}_j|}{|\overline{S}_i \cup \overline{S}_j|},$$

where $\overline{S}_i$ is the set of users rating the item i.

Edges with probability equal to 1, and with label $\mathtt{r}_k$ between the user u and the item i, denoting the user u has rated the item i with a score equal to k, are added for each element r_{ui} belonging to $\mathcal{K}$.

After having defined the uncertain graph, now we can solve classical collaborative filtering task by computing the probability of some language constrained simple paths. Since the goal is to predict an unknown rating between an user u and an item i, let us assume that the values of r_{ui} are discrete and belonging to a set R. Given the uncertain graph G, the approach we used to predict the rating $\widehat{r_{ui}}$ is to solve the following maximization problem:

$$\widehat{r_{ui}} = \arg\max_j P(\mathtt{r}_j(u,i)|G), \tag{8}$$

where $\mathtt{r}_j(u,i)$ is the unknown link with label $\mathtt{r}_j$ between the user u and the item i. In particular, the maximization problem corresponds to compute the link prediction for each rating value and then choosing the rating with maximum likelihood.

The previous link prediction task is based on querying the probability of some language constrained simple path. For instance, user-based CF may be simulated by querying the probability of the paths, starting from a user node and ending to an item node, belonging to the context free language $L_i = \{\mathtt{simU}^1\mathtt{r}_i^1\}$. In particular, predicting the probability of the rating j as $P(\mathtt{r}_j(u,i)$ in (8) corresponds to compute the probability $P(q|G)$ for a query path in L_i, i.e., computing $P(L_i|G)$ as in (3):

$$\widehat{r_{ui}} = \arg\max_j P(\mathtt{r}_j(u,i)|G) \approx \arg\max_j P(L_i|G). \tag{9}$$

In the same way, item-base CF could be simulated by computing the probability of the paths belonging to the CFL $L_i = \{\mathtt{r}_i^1\mathtt{simI}^1\}$.

The power of the proposed framework gives us the possibility to construct more complex queries such as that belonging to the CFL $L_i = \{\mathtt{r}_i\mathtt{simI}^n : 1 \leq n \leq 2\}$, that gives us the possibility to explore the graph by considering not only direct connections. Finally, we can implement hybrid CF systems solving queries belonging to the CFL $L_i = \{\mathtt{r}_i\mathtt{simI}^n : 1 \leq n \leq 2\} \cup \{\mathtt{simU}^m\mathtt{r}_i^1 : 1 \leq m \leq 2\}$.

4 Experiments

In order to validate the proposed approach two versions of the MovieLens[2] dataset has been used. The MovieLens data sets were collected by the GroupLens Research Project at the University of Minnesota. The first version called MovieLens 100K consists of 100,000 ratings (1-5) from 943 users on 1682 movies, where each user has rated at least 20 movies and there are simple demographic info for the users (age, gender, occupation, zip). The data was collected through the MovieLens web site during the seven-month period from September 19th, 1997 through April 22nd, 1998. The second version called MovieLens 1M consists

[2] http://www.grouplens.org/

of 1,000,209 anonymous ratings of approximately 3,900 movies made by 6,040 MovieLens users. In this paper we used the ratings only without considering the demographic information.

MovieLens 100K dataset is divided in 5 fold, where each fold present a training data (80000 ratings) and a test data (20000 ratings), while MovieLens 1M is divided in 10 fold. For each training/testing fold the validation procedure follows the following steps:

1. creating the uncertain graph from the training ratings data set as reported Section 3;
2. defining a context free language corresponding to a specific CF task;
3. testing the ratings reported in the testing data set $\mathcal{T}$ by computing, for each pair $(u, i) \in \mathcal{T}$ the predicted rating as in (9) and comparing the result with the true prediction reported in $\mathcal{T}$.

In this particular dataset we have a uncertain graph with nodes labeled as `user` or as `film`. There are edges between two `film` nodes labeled as `simF`, and there are edges with label `simU` between two `user` nodes. These edges are added using the procedure presented in the previous section, where we set the parameter $n = 30$, indicating that an user or a film is connected, respectively, to 30 most similar users, resp. films . Finally, for each rating $(u, i, r_{ui} = k)$ belonging to the training set there is an edge between the user u and the film i whose label is $\mathtt{r}_k$. The goal is to predict the correct rating for each instance belonging to the testing set $\mathcal{T}$. The predicted rating has been computed using a Monte Carlo approach by sampling 100 certain graphs and adopting the function reported in (9).

The accuracy of the proposed framework has been evaluated according to the *mean absolute error* (MAE) a most commonly applied evaluation metric for CF rating predictions. Assuming N computed rating predictions:

$$MAE = \frac{1}{N} \sum_{i=1}^{N} |\widehat{r_{ui}} - r_{ui}|. \tag{10}$$

4.1 Results

In order to evaluate the framework, we proposed to query the paths belonging to the context free languages reported in Table 1. The first language constrained simple paths L_1 reported in Table 1 corresponds to solve a user-based CF problem, while the second language L_2 gives us the possibility to simulate a item-based CF approach. As we can see from Table 2 results improve when we go from a user-based approach to a item-based one.

Then we try to build a basic hybrid system by combining both the languages L_1 and L_2 into the language L_3. Now, as we can see in Table 2 results are better than that obtained when we used a single language only. Then, we propose to extend the basic languages L_1 and L_2 in order to consider a neighbourhood with many nested levels. In particular, instead of considering the direct neighbours

$L_1 = \{\texttt{simU}^1 \texttt{r}_k^1\}$
$L_2 = \{\texttt{r}_k^1 \texttt{simF}^1\}$
$L_3 = \{\texttt{simU}^1 \texttt{r}_k^1\} \cup \{\texttt{r}_k^1 \texttt{simF}^1\}$
$L_4 = \{\texttt{simU}^n \texttt{r}_k^1 : 1 \leq n \leq 2\}$
$L_5 = \{\texttt{r}_k^1 \texttt{simF}^n : 1 \leq n \leq 2\}$
$L_6 = \{\texttt{r}_k^1 \texttt{simF}^n : 1 \leq n \leq 3\}$
$L_7 = \{\texttt{simU}^n \texttt{r}_k^1 : 1 \leq n \leq 2\} \cup \{\texttt{r}_k^1 \texttt{simF}^n : 1 \leq n \leq 2\}$
$L_8 = \{\texttt{r}_k^1 \texttt{simF}^n : 1 \leq n \leq 4\}$

Table 1. Language constrained simple paths used for the MovieLens dataset.

only, we inspect the uncertain graph following a path with a maximum length of two edges, labeled respectively as `simU` for the language L_4 and `simF` for the language L_5. Their corresponding results are better than that obtained with the basic language L_1 and L_2 thus proving the validity of the approach. Language L_6 extends language L_5 in order to inspect the uncertain graph following a path with a maximum length of three edges by obtaining better results than others languages.

Finally, the language L_7 combines both the user-based and item-based approach, and the large neighbourhood explored with paths whose length is greater than one. As we can see, this language is the best among all the others in providing a good MAE value.

	Path						
Fold	L_1	L_2	L_3	L_4	L_5	L_6	L_7
1	0.9419	0.8458	0.8228	0.8661	0.7928	0.7837	0.7663
2	0.9337	0.8366	0.8119	0.8513	0.7777	0.7800	0.7670
3	0.9189	0.8141	0.8063	0.8505	0.7739	0.7700	0.7584
4	0.9275	0.8273	0.8096	0.8608	0.7784	0.7724	0.7678
5	0.9528	0.8421	0.8312	0.8637	0.7824	0.7754	0.7785
Mean	0.9349	0.8332	0.8164	0.8585	0.7810	0.7763	0.7676

Table 2. MAE error on MovieLens 100K adopting different path type

Table 3 shows the results on the MovieLens 1M dataset, using a 10-fold cross-validation, comparing the proposed framework with respect to a neighborhood-based recommendation method [4] adopting as similarity weight the Mean Squared Difference (MSD), the Spearman Rank Correlation (SRC) or the Pearson Correlation (PC). In this case we adopted another language, L_8, that extends the neighborhood of the explored graph. As we can see, the obtained results adopting our system are better than those obtained with the neighborhood-based approach. Furthermore, more the portion of the explored graph is considered, adopting the languages L_2, L_5, L_6 and L_8, and more is the predictive accuracy reached by the system.

Method	MAE
MSD	0.7602
SRC	0.7529
PC	0.7518
L_2	0.7916
L_5	0.7381
L_6	0.7293
L_8	0.7198

Table 3. MSE error on MovieLens 1M

5 Related works

Given a snapshot of a graph (network), the goal we are dealing with is to accurately predict edges that could be added to the network in future, sometime called *link prediction problem* [5]. There are a lot of application where link prediction can be used such as identifying the structure of a criminal network, overcoming the data-sparsity problem in recommender systems using collaborative filtering [25], analyzing users navigation history to generate users tools that increase navigational efficiency [26]. A problem close to link prediction is link completion [8]. The data, collected from the real life sources, is usually noisy and might contain gaps, i.e. links may be incomplete, containing one or more unknown members. The problem of link completion addresses the task of determining the missing member given a partial link. This question is similar to those found in the collaborative filtering domain [2]. The link prediction problem is also related to the problem of inferring missing links from an observed network: in a number of domains, one constructs a network of interactions based on observable data and then tries to infer additional links that, while not directly visible, are likely to exist [7, 18, 22].

All these methods assign a connection weight $score(x, y)$ or a similarity $s(x, y)$ to pairs of nodes x, y, based on the input graph, and then produce a ranked list in decreasing order of $s(x, y)$. This approach may be viewed as computing a measure of proximity or a similarity between nodes. The most basic approach to compute this ranked list could be that to rank pairs x, y by the length of their shortest path in the network G . Such a measure follows the notion that collaboration networks are *small worlds*, in which individuals are related through short chains [17]. Shortest path between two nodes defines the minimum number of edges connecting them. If there is no such connecting path then, the value of this attribute is taken as infinite.

Other methods try to compute the similarity between two nodes by looking their corresponding neighborhoods. Given a node x, let $N(x)$ be the set of neighbours of x in a graph G. Given two nodes x and y, there are several approaches that follow the natural intuition that if the set of neighbours $N(x)$ and $N(y)$ have a large overlapping then the node x and the node y should be very similar.

Common neighbours measure the number of neighbors that node x and node y have in common, in particular $s(x,y) = |N(x) \cap N(y)|$. Newman in [16] shows a correlation between the number of common neighbours of x and y at the time t, and the probability they will be similar in the future.

Jaccard's coefficient, used in information retrieval, measures the probability that both x and y have a feature f in common, for a randomly selected feature f. Using neighbours we can compute this as follow $s(x,y) = |N(x) \cap N(y)|/|N(x) \cup N(y)|$. [1] considers the similarity problem between two entities as $s(x,y) = \sum_{z \in N(x) \cap N(y)} \frac{1}{log|N(z)|}$ where z is a set of features shared both by x and y. Finally, *preferential attachment* is based on empirical evidence that the probability of x and y being connected is correlated with the product of the number of connections of x and y ($N(x)$ and $N(y)$). The measure is computed as $s(x,y) = |N(x)| \cdot |N(y)|$.

Other methods are based on ensemble of paths. Katz [13] defines a similarity measure that directly sums over a collection of paths, exponentially damped by length in order to count short paths more heavily. This leads to the measure $s(x,y) = \sum_{l=1}^{\infty} \beta^l \cdot |paths_{x,y}^{\langle l \rangle}|$ where $paths_{x,y}^{\langle l \rangle}$ is the set of all lengh-l paths from x to y. There exists two variants of the Katz measure: unweighted, in witch $paths_{x,y}^{\langle 1 \rangle} = 1$ if x and y have collaborated and 0 otherwise, and weighted, in witch $paths_{x,y}^{\langle 1 \rangle}$ is the number of times that x and y have collaborated.

Another method uses random walks on the graph G [23], where starting from a node x, the selection of next node to visit is done by choosing among the neighbors of x at random. Using this approach it is possible to compute the hitting time $H_{x,y}$ as the expected number of steps required for a random walk starting at x to reach y. SimRank [10] supposes that two nodes are similar to the extent that they are joined to similar neighbors. In particular $s(x,y) = \gamma \cdot \frac{\sum_{a \in N(x)} \sum_{b \in N(y)} s(a,b)}{|N(x)| \cdot |N(y)|}$ for some $\gamma \in [0,1]$.

All the methods described above consider the space of representation as a graph with nodes of the network indicating the objects of the world and edges with a numeric value that indicates their weight. Over the last few years uncertain graphs have become an important research topic [19, 27, 28]. In these graphs each edge is associated with an edge existence probability that quantifies the likelihood that the edge exists in the graphs. Using this representation it is possible to adopt the *possible world* semantics to model it. One of main issue in uncertain graphs is how to compute the connectivity of the network. The *network reliability problem* [3] is a generalization of *pairwise reachability*, in which the goal is to determine the probability that all pairs of nodes are reachable from one another. Unlike a deterministic graph in which the reachability function is a binary function indicating whether or not there is a path that connects the two provided vertices, in the case of the reachability on uncertain graphs the function assumes probabilistic values. In [19], the authors provide a list of alternative shortest-path distance measures for uncertain graphs in order to discover the k closest vertices to a given vertex. Another work [12] try to deal with the concept of $x - y$ distance-constraint reachability problem. In particular, given two vertices x and y, they try to solve the problem of computing the probability that

the distance from x to y is less than or equal to a user-defined threshold. In order to solve this problem, they proposed an exact algorithm and two reachability estimators based on probability sampling.

6 Conclusions

In this paper a framework based on uncertain graphs able to deal with collaborative filtering problems has been presented. The evaluation of the proposed approach has been reported by applying it to a real world dataset and proving its validity in solving simple and complex collaborative filtering tasks. As future development we will conduct further experiments in order to accurately validate the framework. We will study how the size of the neighbourhood of each node, during the graph construction phase, could influence the quality of the predictions.

References

1. Adamic, L.A., Adar, E.: Friends and neighbors on the web. Social Networks 25(3), 211–230 (2003)
2. Breese, J.S., Heckerman, D., Kadie, C.: Empirical analysis of predictive algorithms for collaborative filtering. In: Proceedings of the14th Annual Conference on Uncertainty in Artificial Intelligence (UAI98). pp. 43–52 (1998)
3. Colbourn, C.J.: The Combinatorics of Network Reliability. Oxford University Press (1987)
4. Desrosiers, C., Karypis, G.: A comprehensive survey of neighborhood-based recommendation methods. In: Ricci, F., Rokach, L., Shapira, B., Kantor, P.B. (eds.) Recommender Systems Handbook, pp. 107–144. Springer (2011)
5. Getoor, L., Diehl, C.P.: Link mining: a survey. SIGKDD Explorations 7(2), 3–12 (2005)
6. Getoor, L., Taskar, B.: Introduction to Statistical Relational Learning (Adaptive Computation and Machine Learning). The MIT Press (2007)
7. Goldberg, D.S., Roth, F.P.: Assessing experimentally derived interactions in a small world. Proceedings of the National Academy of Sciences 100(8), 4372–4376 (2003)
8. Goldenberg, A., Kubica, J., Komarek, P., Moore, A., Schneider, J.: A comparison of statistical and machine learning algorithms on the task of link completion. In: KDD Workshop on Link Analysis for Detecting Complex Behavior (2003)
9. Hintsanen, P., Toivonen, H.: Finding reliable subgraphs from large probabilistic graphs. Data Mining and Knowledge Discovery 17(1), 3–23 (2008)
10. Jeh, G., Widom, J.: Simrank: a measure of structural-context similarity. In: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 538–543. KDD '02, ACM (2002)
11. Jin, R., Liu, L., Aggarwal, C.C.: Discovering highly reliable subgraphs in uncertain graphs. In: Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 992–1000. KDD '11, ACM, New York, NY, USA (2011)
12. Jin, R., Liu, L., Ding, B., Wang, H.: Distance-constraint reachability computation in uncertain graphs. Proc. VLDB Endow. 4, 551–562 (2011)

13. Katz, L.: A new status index derived from sociometric analysis. Psychometrika 18(1), 39–43 (1953)
14. Koller, D., Friedman, N.: Probabilistic Graphical Models: Principles and Techniques. MIT Press (2009)
15. Koren, Y., Bell, R.M.: Advances in collaborative filtering. In: Ricci, F., Rokach, L., Shapira, B., Kantor, P.B. (eds.) Recommender Systems Handbook, pp. 145–186. Springer (2011)
16. Newman, M.E.J.: Clustering and preferential attachment in growing networks. Phys. Rev. E 64 (2001)
17. Newman, M.E.J.: The structure of scientific collaboration networks. Proceedings of the National Academy of Sciences of the United States of America 98(2), 404–409 (2001)
18. Popescul, A., Ungar, L.H.: Statistical relational learning for link prediction. In: IJCAI03 Workshop on Learning Statistical Models from Relational Data (2003)
19. Potamias, M., Bonchi, F., Gionis, A., Kollios, G.: k-nearest neighbors in uncertain graphs. Proc. VLDB Endow. 3, 997–1008 (2010)
20. Sachan, M., Ichise, R.: Using semantic information to improve link prediction results in network datasets. IACSIT International Journal of Engineering and Technology 2(4), 71–76 (2010)
21. Taranto, C., Di Mauro, N., Esposito, F.: Probabilistic inference over image networks. Italian Research Conference on Digital Libraries 2011 CCIS 249, 1–13 (2011)
22. Taskar, B., Wong, M.F., Abbeel, P., Koller, D.: Link prediction in relational data. In: in Neural Information Processing Systems (2003)
23. von Luxburg, U., Radl, A., Hein, M.: Hitting and commute times in large graphs are often misleading. CORR (2011)
24. Witsenburg, T., Blockeel, H.: Improving the accuracy of similarity measures by using link information. In: Kryszkiewicz, M., Rybinski, H., Skowron, A., Ras, Z.W. (eds.) ISMIS. Lecture Notes in Computer Science, vol. 6804, pp. 501–512. Springer (2011)
25. Zan, H., Xin, L., Hsinchun, C.: Link prediction approach to collaborative filtering. In: Proceedings of the 5th ACM/IEEE-CS joint conference on Digital libraries. pp. 141–142. ACM Press (2005)
26. Zhu, J.: Mining web site link structures for adaptive web site navigation and search. In: PhD thesis. University of Ulster (2003)
27. Zou, Z., Gao, H., Li, J.: Discovering frequent subgraphs over uncertain graph databases under probabilistic semantics. In: Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 633–642. ACM (2010)
28. Zou, Z., Li, J., Gao, H., Zhang, S.: Finding top-k maximal cliques in an uncertain graph. International Conference on Data Engineering pp. 649–652 (2010)
29. Zou, Z., Li, J., Gao, H., Zhang, S.: Mining frequent subgraph patterns from uncertain graph data. IEEE Transactions on Knowledge and Data Engineering 22, 1203–1218 (2010)

Movie Recommendation with DBpedia

Roberto Mirizzi[1], Tommaso Di Noia[1], Azzurra Ragone[2], Vito Claudio Ostuni[1], Eugenio Di Sciascio[1]

[1] Politecnico di Bari – Via Orabona, 4, 70125 Bari, Italy
{mirizzi,ostuni}@deemail.poliba.it, {t.dinoia,disciacio}@poliba.it
[2] Exprivia S.p.A. – Viale A. Olivetti, 11/A, 70056 Molfetta (BA), Italy
azzurra.ragone@exprivia.it

Abstract. In this paper we present MORE (acronym of ***MORE*** *than* ***MO****vie* ***RE****commendation*), a Facebook application that semantically recommends movies to the user leveraging the knowledge within Linked Data and the information elicited from her profile. MORE exploits the power of social knowledge bases (e.g. DBpedia) to detect semantic similarities among movies. These similarities are computed by a Semantic version of the classical Vector Space Model (sVSM), applied to semantic datasets. Precision and recall experiments prove the validity of our approach for movie recommendation. MORE is freely available as a Facebook application.

1 Introduction

The field of recommender systems, from an Information Retrieval (IR) perspective, is in its maturity stage and many applications are available on the Web that recommend items to the end user based on a combination of content-based, collaborative filtering and knowledge-based approaches [16]. In this paper we present MORE[3]: a **movie recommender system** in the Web of Data. Currently, the system relies on one of the most relevant datasets in the Linked Data [3] cloud: DBpedia [4], and on the semantic-enabled version of the *Internet Movie Database* (IMDB): LinkedMDB [11]. It is developed as a Facebook application and uses also a faceted-browsing approach to metadata navigation and exploration. MORE basically exploits the information coming from Linked Data datasets to compute a semantic similarity between movies and provide a recommendation to the user. Since MORE has been implemented as a Facebook application, in order to avoid the *cold start* problem typical of content-based recommender systems, when the user starts using it, we may retrieve information about the movies she likes by grabbing them from her Facebook profile. We use semantic information contained in the RDF datasets to compute a semantic similarity between movies the user might be interested in.

Main contributions of this paper are: (i) presentation of a Facebook application for movie recommendation exploiting semantic datasets; (ii) a Semantic-based Vector Space Model for recommendation of items in Linked Data datasets; (iii) evaluation and validation of the approach with MovieLens dataset.

[3] http://apps.facebook.com/movie-recommendation/

The remainder of the paper is structured as follows: in Section 2 we illustrate how we exploit semantic information contained in RDF datasets to compute semantic similarities between movies, then in Section 3 we describe the interface of MORE. In Section 4 we show how to compute similarities between movies using a semantic-adaptation of the Vector Space Model (VSM). Section 5 introduces the recommender system we developed for Linked Data data while Section 6 shows the results of our evaluation. In Section 7 we review relevant related work. Conclusion and future work close the paper.

2 Social knowledge bases for similarity detection

By exploiting its SPARQL endpoint[4], it is possible to ask complex queries to DBpedia with high precision in the results. For example, we may retrieve which are the movies where *Al Pacino* and *Robert De Niro* starred together, and discover that *Righteous Kill*[5] and *Heat*[6] are two of these movies. Intuitively, we assume that these movies are related with each other, since they share part of the cast. Via SPARQL queries, we may also find that there are other characteristics shared between the two movies, such as some categories (e.g. *crime films*). Roughly speaking, the more features two movies have in common, the more they are similar. In a few words, a similarity between two movies (or two resources in general) can be detected if in the RDF graph:

- they are directly related: this happens for example if a movie is the sequel of another movie. In DBpedia this state is handled by the properties dbpedia-owl:subsequentWork and dbpedia-owl:previousWork.
- they are the subject of two RDF triples having the same property and the same object, as for example when two movies have the same director. In the movie domain, we take into account about 20 properties, such as dbpedia-owl:starring and dbpedia-owl:director. They have been automatically extracted via SPARQL queries. The property dcterms:subject needs a dedicated discussion, as we will see in the following.
- they are the object of two RDF triples having the same property and the same subject.

Categories and genres. Categories in Wikipedia are used to organize the entire project, and help to give a structure to the whole project by grouping together pages on the same subject. The sub-categorization feature makes it possible to organize categories into tree-like structures to help the navigation of the categories. In DBpedia, the hierarchical structure of the categories is modeled through two distinct properties, dcterms:subject and skos:broader. The former relates a resource (e.g. a movie) to its categories, while the latter is used to relate a category to its parent categories. Hence, the similarity between two

[4] http://dbpedia.org/sparql
[5] http://en.wikipedia.org/wiki/Righteous_Kill
[6] http://en.wikipedia.org/wiki/Heat_(1995_film)

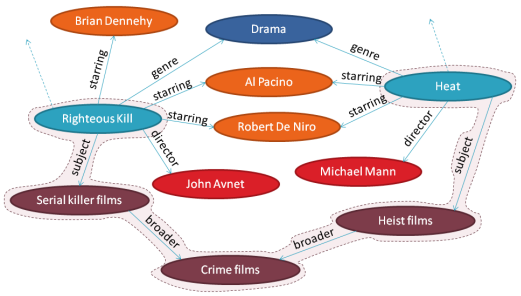


Fig. 1. A sample of an RDF graph related to the movie domain.

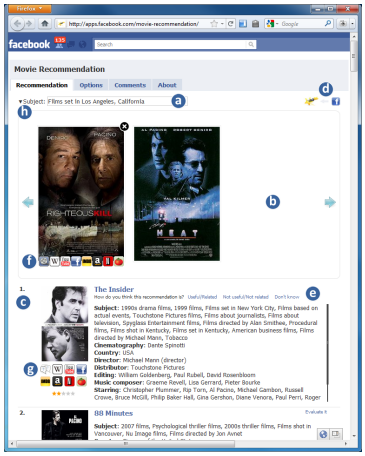


Fig. 2. A screenshot of MORE.

movies can be also discovered in case they have some ancestor categories in common (within the hierarchy). This allows one to catch implicit relations and hidden information, i.e. information that is not directly detectable looking only at the nearest neighbors in the RDF graph. As an example, thanks to the categories, it is possible to infer a relation between *Righteous Kill* and *Heat*, since they both belong (indirectly) to the *Crime films* category, as shown with the highlighted path in Fig. 1, which shows a sample of the RDF graph containing properties and resources coming both from DBpedia and from LinkedMDB/IMDB..

3 MORE: More than Movie Recommendation

In this section we describe MORE, our Facebook application for movie recommendation. A screenshot of the application is depicted in Fig. 2. Although the application exploits semantic datasets, the complex semantic nature of the underlying information is hidden to the end user. She does not interact directly with Semantic Web languages and technologies such as RDF and SPARQL. Despite the choice of the movie domain, we stress that, since our system relies on semantic knowledge bases, it is potentially able to generate recommendations for any areas covered by DBpedia and, more generally, for any dataset in the Linked Data cloud.

After the application is loaded, the user may search for a *movie* by typing some characters in the corresponding text field, as indicated by *(a)* in Fig. 2. The system returns an *auto-complete list* of suggested movies, ranked by popularity in DBpedia. In order to rank the movies in the *auto-complete list*, we adapted the PageRank algorithm to the DBpedia subgraph related to movies. To this aim we consider the property dbpedia-owl:wikiPageWikiLink which corresponds to links between Wikipedia pages. In ranking the results shown in the auto-complete list, we consider also non-topological information by weighting the

results coming from the previous computation with votes on movies from IMDB users.

Once the list has been populated, the user can select one of the suggested movies. Then, the chosen movie is placed in the user's favorite movies area (see *(b)* in Fig. 2) and a **recommendation** of the top-40 movies related to the selected one is presented to the user (see *(c)* in Fig. 2). The relevance rankings for the movies are computed (off-line) as detailed in Section 4. The user can add more movies to her favorite list, just clicking either on its poster or on its title appearing in the recommendation list. Then, the movie is moved into the favorite area and the recommendation list is updated taking into account also the item just added. Another way to add a movie to the favorite list is to exploit the functionalities offered by the Facebook platform and the Graph API[7]. Facebook users can add their favorite movies to their own Facebook profile. In MORE, the user can obtain her preferred Facebook movies by clicking on the icon indicated with *(d)* in Fig. 2. Then, the user can select a movie from the returned list, in order to add it to the favorite area and to obtain the related recommendation. Each of these actions are tracked by the system. In fact, our long run goal is to collect relevant information about user preferences in order to provide a personalized recommendation that exploits both the knowledge bases such as DBpedia or LinkedMDB (**content-based** approach) and the similarities among users (**collaborative-filtering** approach). The user is allowed to set her personal preferences about the properties involved in the recommendation using the sliders in the *Options* tab. In Section 4 we will detail how we automatically compute a default value for the weights associated to each property.

4 Semantic Vector Space Model

In order to compute the similarities between movies, we propose a semantic-adaptation of one of the most popular models in classic information retrieval [1]: the Vector Space Model (VSM) [17]. In VSM non-binary weights are assigned to index terms in queries and in documents (represented as sets of terms), and are used to compute the degree of similarity between each document in the collection and the query. In our approach, we semanticized the classical VSM, usually used for text retrieval, to deal with RDF graphs. In a nutshell, we represent the whole RDF graph as a 3-dimensional tensor where each slice refers to an ontology property. Given a property, each movie is seen as a vector, whose components refer to the *term frequency-inverse document frequency* TF-IDF (or better, in this case, *resource frequency-inverse movie frequency*). For a given slice (i.e. a particular property), the similarity degree between two movies is the correlation between the two vectors, and it is quantified by the cosine of the angle between them. An RDF graph can be viewed as a labeled graph $G = (V, E)$, where V is the set of RDF nodes and E is the set of predicates (or properties) between nodes in V. In our model, an RDF graph is then a 3-dimensional tensor $\mathbf{T}$ where each slice identifies an adjacency matrix for an RDF property (see Fig. 3). All the nodes in

[7] http://developers.facebook.com/docs/reference/api/

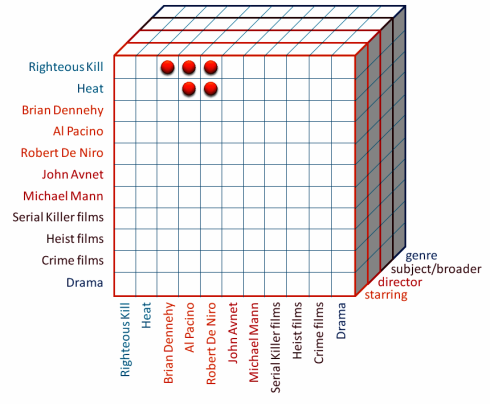


Fig. 3. Tensor representation of the RDF graph of Fig. 1. Only the components on the first slice (i.e. *starring*) are visible.

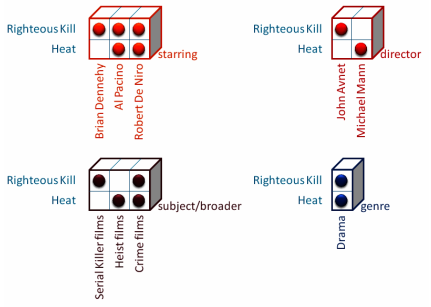


Fig. 4. Slices decomposition.

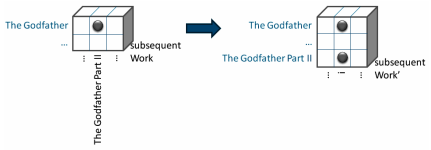


Fig. 5. Property transformation.

V are represented both on the rows and on the columns. A component (i.e. a cell in the tensor) is not *null* if there is a property that relates a subject (on the rows) to an object (on the columns). A few words need to be spent for the properties `dcterms:subject` and `skos:broader`. As also shown in Fig. 1 every movie is related to a category by the property `dcterms:subject` which is in turn related to other categories via `skos:broader` organized in a hierarchical structure. In order to catch such a relation, for each resource we computed the transitive closure of the category it is related to and assign the whole set of computed categories as the value of `dcterms:subject` of the corresponding movie. As an example, going back to the small example depicted in Fig. 1, the set of values assigned to `dcterms:subject` for *Righteous Kill* is {*Serial Killer films*, *Crime Films*}. This can be viewed as an explicit representation of the two following triples:

```
dbpedia:Righteous_Kill dcterms:subject dbpedia:Category:Serial_killer_films
dbpedia:Righteous_Kill dcterms:subject dbpedia:Category:Crime_films
```

Looking at the model, we may observe and remember that: (1) the tensor is very sparse; (2) we consider properties as independent with each other (there is no `rdfs:subPropertyOf` relation); (3) we are interested in discovering the similarities between movies (or in general between resources of the same `rdf:type` and not between any pair of resources). Based on the above observations, we can decompose the tensor slices into smaller matrices. Each matrix of Fig. 4 refers to a specific RDF property, and corresponds to a slice in the tensor. In other words, for each matrix, the rows represent somehow the *domain* of the considered property, while the columns its *range*. For a given property, the components of each

row represent the contribution of a resource (i.e. an actor, a director, etc.) to the corresponding movie. With respect to a selected property p, a movie m is then represented by a vector containing all the terms/nodes related to m via p. As for classical Information Retrieval, the index terms $k_{n,p}$, that is all the nodes n linked to a movie by a specific property p, are assumed to be all mutually independent and are represented as unit vectors of a t-dimensional space, where t is the total number of index terms. Referring to Fig. 4, the index terms for the *starring* property are *Brian Dennehy*, *Al Pacino* and *Robert De Niro*, while $t = 3$ is the number of all the actors that are objects of a triple involving *starring*. The representation of a movie m_i, according to the property p, is a t-dimensional vector given by:

$$\overrightarrow{m_{i,p}} = (w_{1,i,p}, w_{2,i,p}, ..., w_{t,i,p})$$

where $w_{n,i,p}$ is a non-negative and non-binary value representing the weight associated with a term-movie pair $(k_{n,p}, \overrightarrow{m_{i,p}})$. The weights $w_{n,i,p}$ we adopt in our model are TF-IDF weights. More precisely they are computed as:

$$w_{n,i,p} = f_{n,i,p} * \log\left(\frac{M}{a_{n,p}}\right)$$

where $f_{n,i,p}$ represents the TF, i.e. the frequency of the node n, as the object of an `RDF` triple having p as property and the node i as subject (the movie). Actually, this term can be at most 1, since two identical triples can not coexist in an `RDF` graph. Then, in case there is a triple that links a node i to a node n via the property p, the frequency $f_{n,i,p}$ is 1, otherwise $f_{n,i,p} = 0$, and the corresponding weight $w_{n,i,p}$ is set to 0. M is the total number of movies in the collection, and $a_{n,p}$ is the number of movies that are linked to the resource n, by means of the predicate p. As an example, referring to Fig. 4, for the *starring* property, and considering $n = AlPacino$, then $a_{AlPacino,starring}$ is equal to 2, and it represents the number of movies where *Al Pacino* acted. Relying on the model presented above, each movie can be represented as a $t \times P$ matrix (it corresponds to a horizontal slice in Fig. 3), where P is the total number of selected properties. If we consider a projection on a property p, each pair of movies, m_i and m_j, are represented as t-dimensional vectors. As for classical VSM, here we evaluate the degree of similarity of m_i with respect to m_j, as the correlation between the vectors $\overrightarrow{m_i}$ and $\overrightarrow{m_j}$. More precisely we calculate the cosine of the angle between the two vectors as:

$$sim^p(m_i, m_j) = \frac{\overrightarrow{m_{i,p}} \bullet \overrightarrow{m_{j,p}}}{|\overrightarrow{m_{i,p}}| \times |\overrightarrow{m_{j,p}}|} = \frac{\sum_{n=1}^{t} w_{n,i,p} \cdot w_{n,j,p}}{\sqrt{\sum_{n=1}^{t} w_{n,i,p}^2} \cdot \sqrt{\sum_{n=1}^{t} w_{n,j,p}^2}}$$

Such a value is the building block of our content-based recommender system. By means of the computed similarities, it is possible to ask the system questions like *"Which are the most similar movies to movie m_i according to the specific property $\tilde{p}$?"*, and also "*Which are the most similar movies to movie m_i according to the whole knowledge base?*". In the following we will see how to combine such values with a user profile to compute a content-based recommendation.

5 Semantic content-based Recommender System

The method described so far is general enough and it can be applied when the similarity has to be found between resources that appear as subjects or object of RDF triples[8]. Another case is about how to discover a similarity between resources that are directly related by some specific properties. In the considered movie domain, this situation happens for example with the *subsequentWork* property. In our approach we operate a matrix transformation to revert this situation to the one considered so far. The transformation is illustrated in Fig. 5. In order to use the VSM with two resources directly linked, the property p is transformed into the property p' and its domain remains unchanged. The object of the original RDF triple for the new property p' is mapped into a unique index associated to the original object (in Fig. 5, index i is associated with *The Godfather Part II*), and a new RDF triple is created having as subject the original object and as object the index just created. Referring to Fig. 5, *The Godfather Part II* becomes the subject of a new triple, where the predicate is *subsequentWork'* and the object is the index i. Now our semantic VSM can be applied straight.

If we want to provide an answer also to questions like "*Which are the most similar movies to movie m_i according to the user profile?*" we need a step further to represent the user profile. In our setting, we model it based on the knowledge we have about the set of rated movies. In MORE we have information on the movies the user likes. Hence, the profile of the user u is the set:

$$profile(u) = \{m_j \mid u \text{ likes } m_j\}$$

In order to evaluate if a new resource (movie) m_i might be of interest for u — with $m_i \notin profile(u)$ — we compute a similarity $\tilde{r}(u, m_i)$ between m_i and the information encoded in $profile(u)$ via Equation (1).

$$\tilde{r}(u, m_i) = \frac{\sum_{m_j \in profile(u)} \frac{1}{P} \sum_p \alpha_p \cdot sim^p(m_j, m_i)}{|profile(u)|} \tag{1}$$

In Equation (1) we use P to represent the number of properties we selected (see Section 4) and $|profile(u)|$ for the cardinality of the set $profile(u)$. The formula we adopted to compute $\tilde{r}(u, m_i)$ takes into account the similarities between the corresponding properties of the new item m_i and $m_j \in profile(u)$. A weight α_p is assigned to each property representing its worth with respect to the user profile. If $\tilde{r}(u, m_i) \geq 0.5$ then we suggest m_i to u. We want to stress here that, as discussed in the next section, setting a threshold different from 0.5 does not affect the system results.

[8] When the resources to be ranked appear as objects of RDF triples, it is simply a matter of swapping the rows with the columns in the matrices of Fig. 4 and applying again the same algorithm.

	$\alpha_{subject}$	$\alpha_{director}$	α_{writer}	$\alpha_{starring}$	$error$
α^1	0.123	0.039	0.080	0.159	3
α^2	0.024	0.061	0.274	0.433	5
α^3	0.267	0.356	0.188	0.099	3
α^4	0.494	0.428	0.244	0.230	4
α^5	0.082	0.457	0.484	0.051	1

Table 1. Example of values computed after the training.

5.1 Training the system

Although `MORE` allows the user to manually set a value for each α_p, the system automatically computes their default value by training the model via a genetic algorithm. Similarly to an *N-fold cross validation* [16], we split $profile(u)$ in *five* disjoint sets and used alternatively each of them as a *validation set* and the items in the remaining sets as the *training set* of the genetic algorithm. We selected N = 5 because, based on our experimental evaluation, it represents a good trade-off between computational time and accuracy of results. As a matter of fact, every time a new movie is added to $profile(u)$, we re-compute the values of α_p related to u and train again the model for N times. Hence, the higher is N, the more is the time needed to update the result set of the user. During the training step, in order to classify the movies as "*I like*" for u we imposed a threshold of 0.5 for $\tilde{r}(u, m_i)$. It is noteworthy that the threshold can be set arbitrarily since the genetic algorithm computes α_p to fit that value. Hence, if we lower or we raise the threshold the algorithm will compute new values for each α_p according to the new threshold value. After this procedure is completed, we have a set of *five* different values $A_p = \{\alpha_p^1, \ldots, \alpha_p^5\}$ for each α_p. Each value of A_p corresponds to a different round of training. An example of a possible outcome for a small subset of the properties we have in our model is represented in Table 5.1. The last column represents the *misclassification error* computed by the genetic algorithm, i.e., how many resources m_i are not classified as "*I like*" since $\tilde{r}(u, m_i) < 0.5$. Please note that, ideally, the perfect values for α_p would be those returning a misclassification error equal to 0. Indeed, in this step, the movies we consider in our *validation sets* come directly from the user profile. In order to select the best value for each α_p, we considered different options and we tested which one performed better in terms of precision and recall (see Section 6) in the recommendation step. In particular, we evaluated the system performances in the following cases:

$$\alpha_p = \begin{cases} \textbf{min}(\alpha_p^k \in A_p) \\ \textbf{max}\ (\alpha_p^k \in A_p) \\ \textbf{avg}(\alpha_p^k \in A_p) \\ \alpha_p^k \text{ is the } \textbf{median} \text{ of } A_p \\ \alpha_p^k \text{ with the lowest } error \end{cases}$$

The first three options consider an aggregated value computed starting from A_p while the last one consider the tuple with the lower misclassification error. In Figure 6(a) we show how precision and recall of the final recommendation

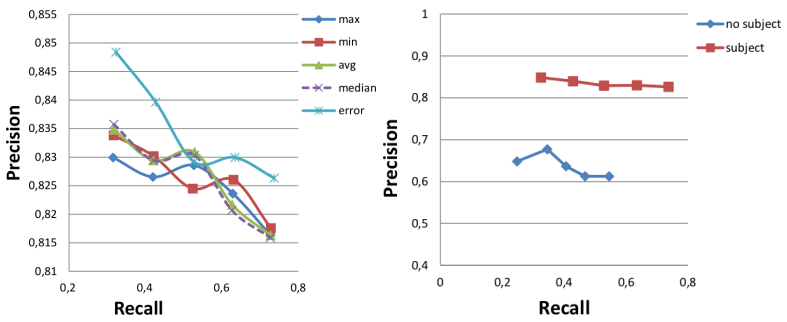


Fig. 6. (a) Precision and recall of the recommendation algorithm with respect to the computation of α_p. (b) Comparison of precision and recall curves with and without `dcterms:subject`.

algorithm vary according to the five cases shown above. We see that the best results are obtained if we consider α_p^k with the lowest misclassification error.

6 Evaluation

In order to evaluate the quality of our algorithm, we performed the evaluation on `MovieLens`, the historical dataset for movie recommender systems. The 100k dataset contains 100,000 ratings from 943 users on 1,682 movies. `MovieLens` datasets are mainly aimed at evaluating collaborative recommender systems in the movie domain. Since our approach is based on a content-based recommendation, in order to use such datasets to test the performances of our algorithms, we linked resources represented in `MovieLens` to `DBpedia` ones. We extracted the value of `rdfs:label` property from all the movies in `DBpedia`, together with the year of production, via `SPARQL` queries. Then, we performed a one-to-one mapping with the movies in `MovieLens` by using the Levenshtein distance and checking the year of production. We found that 78 out of 1,682 (4.64%) movies in `MovieLens` have no correspondence `DBpedia`. After this automatic check we manually double-checked the results and we found that 19 out of 1,604 mappings (1.18%) were not correct and we manually fixed them. Once we had `MovieLens` and `DBpedia` aligned, we tested our pure content-based algorithm by splitting, for each user, the dataset in a *training set* and in a *test set* as provided on the `MovieLens` web-site (80% of the movies rated by the user as belonging to the training set and the remaining 20% as belonging to the *test set*). Before we started our evaluation, we had to align also the user profiles in `MORE` with the ones in `MovieLens`. Indeed, while in more we have only "*I like*" preferences, in `MovieLens` the user u may express a rate on a movie m_j based on a five-valued scale: $r(u, m_j) \in [1, \ldots, 5]$. Hence, following [2] and [15] we build $profile(u)$ as

$$profile(u) = \{m_j \mid r(u, m_j) \in [4, 5]\}$$

In other words, we consider that u *likes* m_j if they rated it with a score greater or equal to 4 and then they are considered as *relevant* to u. The same consideration holds when we evaluate the recommendation algorithm in terms of precision and recall. In recommender systems, precision and recall are defined respectively as: *precision*: fraction of the top-N recommended items that are relevant to u; *recall*: fraction of the relevant items that are recommended to u. In our experiments, since we focus on the test set to find the actual relevant items of the target user, the top-N list we compute only contains items that are in the target user's test set. We varied N in $\{3, 4, 5, 6, 7\}$ and computed the so-called precision@N and recall@N [1]. We did not consider values with N$>$ 7 since in the `MovieLens` dataset we used there are only a few users who rated more than 7 movies as *relevant*. Precision and recall results for `MORE` are shown in Figure 6(a). We also ran our algorithm without taking into account the property `dcterms:subject` in the movie description. The aim of this experiment was to evaluate how important is the ontological information contained in the `DBpedia` categories in the recommendation process. After all, this information can be found only in ontological datasets. In Figure 6(b) we compare precision and recall graphs both when we consider the knowledge carried by `dcterms:subject` and when we do not use it. As we expected, if we do not consider ontological information, the recommendation results get worse drastically.

7 Related Work

`MORE` is intended to be a meeting point between exploratory browsing and content-recommendation in the Semantic Web, exploiting the huge amount of information offered by the Web of Data. Several systems have been proposed in literature that address the problem of movie recommendations, even if there are very few approaches that exploit the `Linked Data` initiative to provide semantic recommendations. In the following we give a brief overview of semantic-based approaches to (movie) recommendation. Szomszor et al. [19] investigate the use of folksonomies to generate tag-clouds that can be used to build better user profiles to enhance the movie recommendation. They use an ontology to integrate both `IMDB` and `Netflix` data. However, they compute similarities among movies taking into account just similarities between movie-tags and keywords in the tag-cloud, without considering other information like actors, directors, writers as we do in `MORE`. *Filmtrust* [9] integrates Semantic web-based social networking into a movie recommender system. Trust has been encoded using the `FOAF` Trust Module and is exploited to provide predictive movie recommendation. It uses a collaborative filtering approach as many other recommender systems, as *MovieLens* [12], *Recommendz* [8] and *Film-Consei* [14]. Our `RDF` graph representation as a three-dimensional tensor has been inspired by [7]. Tous and Delgado [20] use the vector space model to compute similarities between entities for ontology alignment, however with their approach it is possible to handle only a subset of the cases we consider, specifically only the case where resources are directly linked. Eidon et al. [6] represent each concept in an `RDF` graph as a vector con-

taining non-zero weights. However, they take into account only the distance from concepts and the sub-class relation to compute such weights. Effective user interfaces play a crucial role in order to provide a satisfactory user experience during an exploratory search or a content recommendation. Nowadays, there are some initiatives that exploit the Linked Data cloud to provide effective recommendations. One of these is *dbrec* [13], a music content-based recommender system that adopts an algorithm for *Linked Data Semantic Distance*. It uses DBpedia as knowledge base in the Linked Data cloud. The recommendation is link-based, i.e. the "semantics" of relations is not exploited since each relation has the same importance, and it does not take into account the links hierarchy, expressed in DBpedia through the DCTERMS and SKOS vocabulary.

One of the main issues collaborative-filtering recommenders suffer from is the well known *cold-start* problem [18], where no user preference information is known to be exploited for recommendations. In such cases, almost nothing is known about user preferences [10]. Being our system developed as a Facebook application, it is able to automatically extract the favorite movies from the user profile and to provide recommendations also for new users. In [5] the authors propose a hybrid recommender system where user preferences and item features are part of a semantic network. Partially inspired by this work, we offer the capability of inferring new knowledge from the relations defined in the underlying ontology. One of the most complex tasks of their approach is the building of the concepts within the semantic network. Being MORE based on Linked Data and DBpedia, we do not suffer from this problem since it is quite easy to extract, via SPARQL queries, a DBpedia subgraph related to the movie domain.

8 Conclusion and Future Work

The use of Linked Data datasets poses new challenges and issues in the development of next generation systems for recommendation. In this paper we have presented MORE, a Facebook application that works as a recommender system in the movie domain. The background knowledge adopted by MORE comes exclusively from semantic datasets. In particular, in this version of the tool we use DBpedia and LinkedMDB to collect information about movies, actors, directors, etc.. The recommender algorithm relies on a semantic version of the classical Vector Space Model adopted in Information Retrieval. We are willing to better integrate MORE in the Linked Data cloud by publishing our recommendation using the *Recommendation Ontology*[9]. From a methodological perspective, we are collecting information from MORE users to implement also a collaborative-filtering approach to recommendation. This is particularly relevant and challenging since the application is integrated with Facebook.

Acknowledgments. The authors acknowledge partial support of HP IRP 2011. Grant CW267313.

[9] http://purl.org/ontology/rec/core#

References

1. R. A. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval: The Concepts and Technology behind Search*. Addison-Wesley Professional, 2011.
2. C. Basu, H. Hirsh, and W. Cohen. Recommendation as classification: Using social and content-based information in recommendation. In *In Proc. of the 15th National Conf. on Artificial Intelligence*, pages 714–720. AAAI Press, 1998.
3. C. Bizer, T. Heath, and T. Berners-Lee. Linked data - the story so far. *Int. J. Semantic Web Inf. Syst*, 5(3):1–22, 2009.
4. C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann. Dbpedia - a crystallization point for the web of data. *Web Semant.*, 7:154–165, September 2009.
5. I. Cantador, A. Bellogín, and P. Castells. A multilayer ontology-based hybrid recommendation model. *AI Commun.*, 21:203–210, April 2008.
6. Z. Eidoon, N. Yazdani, and F. Oroumchian. A vector based method of ontology matching. In *Proc. of 3rd Int. Conf. on Semantics, Knowledge and Grid*, pages 378–381, 2007.
7. T. Franz, A. Schultz, S. Sizov, and S. Staab. Triplerank: Ranking semantic web data by tensor decomposition. In *Proc. of 8th ISWC*, pages 213–228, 2009.
8. M. Garden and G. Dudek. Semantic feedback for hybrid recommendations in recommendz. In *IEEE Int. Conf. EEE'05*, pages 754–759, 2005.
9. J. Golbeck and J. Hendler. Filmtrust: Movie recommendations using trust in web-based social networks. In *Proceedings of the IEEE CCNC*, 2006.
10. H. Guo. Soap: Live recommendations through social agents. In *5th DELOS Workshop on Filtering and Collaborative Filtering*.
11. O. Hassanzadeh and M. P. Consens. Linked Movie Data Base. In *Proceedings of the WWW2009 Workshop on Linked Data on the Web (LDOW2009)*, April 2009.
12. J. Herlocker, J. A. Konstan, and J. Riedl. Explaining collaborative filtering recommendations. In *Proceeding on the ACM 2000 Conference on Computer Supported Cooperative Work*, pages 241–250, 2000.
13. A. Passant. dbrec: music recommendations using dbpedia. In *Proc. of 9th Int. Sem. Web Conf.*, ISWC'10, pages 209–224, 2010.
14. P. Perny and J. Zucker. Preference-based search and machine learning for collaborative filtering: the film-consei recommender system. *Information, Interaction, Intelligence*, 1:9–48, 2001.
15. A. Rashid, S. Lam, A. LaPitz, G. Karypis, and J. Riedl. Towards a scalable nn cf algorithm: Exploring effective applications of clustering. In *Advances in Web Mining and Web Usage Analysis*, LNCS, pages 147–166. 2007.
16. F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, editors. *Recommender Systems Handbook*. Springer, 2011.
17. G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Commun. ACM*, 18:613–620, November 1975.
18. A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock. Methods and metrics for cold-start recommendations. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '02, pages 253–260. ACM, 2002.
19. M. Szomszor, C. Cattuto, H. Alani, K. O'Hara, A. Baldassarri, V. Loreto, and V. D. Servedio. Folksonomies, the semantic web, and movie recommendation. In *4th European Semantic Web Conference*, 2007.
20. R. Tous and J. Delgado. A vector space model for semantic similarity calculation and owl ontology alignment. In *DEXA*, pages 307–316, 2006.

Cold Start Problem: a Lightweight Approach at ECML/PKDD 2011 - Discovery Challenge

Leo Iaquinta and Giovanni Semeraro

University of Bari "Aldo Moro", v. Orabona 4, 70125 Bari, Italy
{iaquinta, semeraro}@di.uniba.it

Abstract. The paper presents our participation [5] at the ECML/PKDD 2011 - Discovery challenge for the task on the cold start problem. The challenge dataset was gathered from VideoLectures.Net web site that exploits a Recommender System (RS) to guide users during the access to its large multimedia repository of video lectures. Cold start concerns performance issues when new items and new users should be handled by a RS and it is commonly associated with pure collaborative filtering-based RSs. The proposed approach exploits the challenge data to predict the frequencies of pairs of cold items and old items and then the highest values are used to provide recommendations.

1 Background and Motivation

Recommender systems usually suggest items of interest to users by exploiting explicit and implicit feedbacks and preferences, usage patterns, and user or item attributes. Past behaviour is assumed to be useful to make reliable predictions, thus past data is used in the training of RSs to achieve accurate prediction models. A design challenge comes from the dynamism of real-world systems because new items and new users whose behaviour is unknown are continuously added into the system. As a consequence, recommendations may be negatively affected by the well-known *cold start* problem.

Cold start is commonly associated with pure collaborative filtering-based RSs. Particularly, item-based collaborative filtering techniques assume that items are similar when they are similarly rated and therefore the recommendations concern items with the highest correlations according to the usage evidence. A straight drawback is that new items cannot be recommended because there is not an adequate usage evidence.

Prediction involving *cold* items requires different approaches by comparing the performance for the predictions about *hot* items. This may be desirable due to other considerations such as novelty and serendipity. Thus evaluating the system accuracy on cold items it may be wise to consider that there is a trade-off with the entire system accuracy [7].

The first of the two tasks of the ECML/PKDD 2011 - Discovery Challenge[1] was focused on the cold start problem. The used dataset was gathered from VideoLectures.Net web site. Indeed, VideoLectures.Net

[1] http://www.ecmlpkdd2011.org/challenge.php

exploits a RS to guide users during the access to its large multimedia repository of video lectures. The main entities of the dataset are the lectures. They are described by a set of attributes and of relationships. The attributes are of various kind: for instance, *type* can have one value in a predefined set (lecture, keynote, tutorial, invited talk and so on); *views* attribute has a numeric value; *rec_date* and *pub_date* have a date value; *name* and *description* are unstructured text, usually in the same language of the lecture. The relationships link the lectures with 519 context events, 8,092 authors, and 348 categories. Each of these entities has its own attributes and relationships to describe taxonomies of events and categories. The lectures are divided into 6,983 for the training and 1,122 for the testing as cold items.
In addition, the dataset contains records about pairs of lectures viewed together (not necessarily consecutively) with at least two distinct cookie-identified browsers. This kind of data has a collaborative flavour and it is actually the only information about the past behaviour. The user identification is missing, thus any user personalization is eliminated. User queries and feedbacks are also missing.

2 Proposed Approach

To overcome the cold start problem in the approaches based on collaborative filtering, a common solution is to hybridize them with techniques that do not suffer from the same problem [1]. Thus, a content-based approach is used to bridge the gap between existing items and new ones: item attributes are used to infer similarities between items.
The proposed solution is obtained mainly by three steps: the data preprocessing, the model learning, and the recommendation.

Data pre-processing step starts with obtaining an in-memory object-oriented representation of provided data.
The main output of this step is a set of 20 numeric values describing the similarities between lectures of each pair in the training set. The used features involve language, description, recording and publication ages, conference, authors and their affiliations, and categories. More details are reported in [5].

Model learning step allows to obtain a prediction model for the frequency of a pair of lectures. The available data and the lightweight goal determined the selection of a linear model for the learning problem. Used features for different learned models are reported in [5].
The learned weights of a model are stored in a configuration file, with the option to add a boost factor for each weight to easily explore the feature influences beside the learned model. Fig. 1a and Fig. 1b report the values of the evaluation metric (Mean Average R-precision - MARp) for the recommendations using the model with all the available features when a boost factor is changed. Fig. 2 reports the evaluation metric values for the submitted solutions when the boost factors for the learned weight are changed: the submitted solutions always outperform the provided random baseline (MARp: 0.01949).

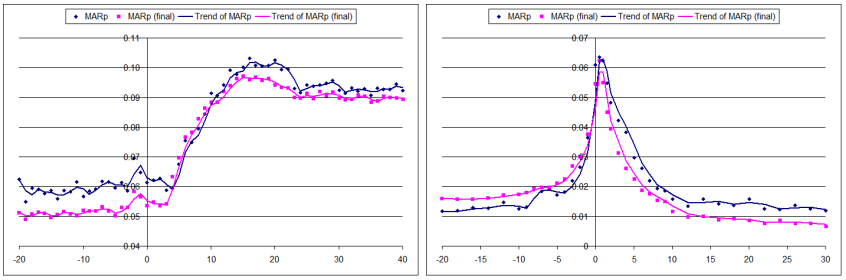


Fig. 1. Boost factor effects for "categoryBest" and "deltaRecAge"

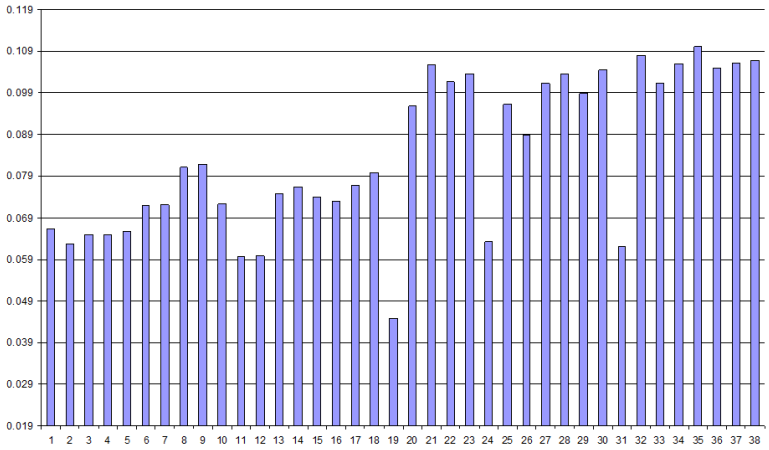


Fig. 2. Mean Average R-precision of submitted solutions

Recommendation step uses the in-memory representation of the pre-processing step and the learned weights to predict the pair frequency of an old item against each selected cold item. The highest values are used to provide recommendations.

2.1 Scale Problem

With the growth of the dataset, many recommendation algorithms are either slowed down or require additional resources such as computation power or memory. As such, it is often the case that algorithms trade other properties, such as accuracy or coverage, for providing rapid results for huge datasets [2]. The trade-off can be achieved by changing some parameters, such as the complexity of the model, or the sample size.
RSs are expected in many cases to provide recommendation on-line, thus it is also important to measure how fast does the system provides recommendation [3, 6]. Common measurement are the number of recommendations that the system can provide per second (the throughput of

the system) and the required time for making a recommendation (the latency or response time).
The developed components allow to complete the recommendation task for the 5,704 lectures in almost 85 seconds on a notebook with an Intel Core 2 at 2.0 GHz as CPU and 2GB of RAM, i.e., each new recommendation about 30 cold items over the selected 1,122 ones is provided in almost 15 milliseconds. Reasonably, a production server allows to reduce further the response time for new recommendations and a cache specifically devised for the recommendations allows to increase the throughput.

3 Conclusions

We have described the steps to achieve the submitted solution that outperforms the random baseline at the ECML/PKDD 2011 - Discovery challenge. The content-based hybrid approach allows to deal the cold start problem. In addition it chances to provide also serendipitous recommendations alongside classical ones [4]. Indeed the content-based item similarity can be used to spot potential serendipitous items as further trade-off with the entire system accuracy.
Finally, the scalability performance is considered as a primary requirement and a lightweight solution is pursued. The preliminary performance for the notebook execution is quite promising and some future directions for improving latency and throughput are sketched.

References

1. Burke, R.: Hybrid recommender systems: Survey and experiments. User Modeling and User-Adapted Interaction 12, 331–370 (2002)
2. Das, A.S., Datar, M., Garg, A., Rajaram, S.: Google news personalization: scalable online collaborative filtering. In: Proc. of the 16th int. conf. on World Wide Web (WWW '07). pp. 271–280. ACM (2007)
3. Herlocker, J., Konstan, J.A., Riedl, J.: An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms. Information Retrieval 5, 287–310 (2002)
4. Iaquinta, L., de Gemmis, M., Lops, P., Semeraro, G., Filannino, M., Molino, P.: Introducing serendipity in a content-based recommender system. In: Xhafa, F., Herrera, F., Abraham, A., Köppen, M., Bénitez, J.M. (eds.) Proc. of the 8th int. conf. on Hybrid Intelligent Systems (HIS-2008). pp. 168–173. IEEE Computer Society (2008)
5. Iaquinta, L., Semeraro, G.: Lightweight approach to the cold start problem in the video lecture recommendation. In: Šmuc, T., Antonov-Fantulin, N., Morzy, M. (eds.) Proc. of the ECML/PKDD Discovery Challenge Workshop. CEUR, vol. 770, pp. 83–94 (2011)
6. Sarwar, B., Karypis, G., Konstan, J., Reidl, J.: Item-based collaborative filtering recommendation algorithms. In: Proc. of the 10th int. conf. on World Wide Web (WWW '01). pp. 285–295. ACM (2001)
7. Shani, G., Gunawardana, A.: Evaluating recommendation systems. In: Ricci, F., Rokach, L., Shapira, B., Kantor, P.B. (eds.) Recommender Systems Handbook, pp. 257–297. Springer (2011)

Comparing Word Sense Disambiguation and Distributional Models for Cross-Language Information Filtering

Cataldo Musto, Fedelucio Narducci, Pierpaolo Basile, Pasquale Lops, Marco de Gemmis, Giovanni Semeraro

Department of Computer Science - University of Bari "Aldo Moro", Italy
{cataldomusto,narducci,basilepp,lops,degemmis,semeraro}@di.uniba.it

Abstract. In this paper we deal with the problem of providing users with cross-language recommendations by comparing two different content-based techniques: the first one relies on a knowledge-based word sense disambiguation algorithm that uses MultiWordNet as sense inventory, while the latter is based on the so-called *distributional hypothesis* and exploits a dimensionality reduction technique called Random Indexing in order to build language-independent user profiles.
This paper summarizes the results already presented within the conference *AI*IA 2011* [1].

Keywords: Cross-language Information Filtering, Word Sense Disambiguation, Distributional Models

1 Introduction

Nowadays the amount of information we have to deal with is usually greater than the amount of information we can process in an effective way. In this context Information Filtering (IF) systems are rapidly emerging since they can adapt their behavior to individual users by learning their preferences and performing a progressive removal of non-relevant content. Specifically, the content-based filtering approach analyzes a set of documents (usually textual descriptions of items) and builds a model of user interests based on the features (usually keywords) that describe the items previously rated as relevant by an individual user. One relevant problem related to content-based approaches is the strict connection with the user language, since the information already stored in the user profile cannot be exploited to provide suggestions for items whose description is provided in other languages. In this paper we investigated whether it is possible to represent user profiles in order to create a mapping between preferences expressed in different languages. Specifically, we compared two approaches: the first one exploits a Word Sense Disambiguation (WSD) technique based on MultiWordnet, while the second one is based on the *distributional models*. It assumes that in every language each term often co-occurs with the same other terms (expressed

in different languages, of course) thus, by representing a content-based user profile in terms of the co-occurences of its terms, user preferences become inerently independent from the language. The paper is organized as follows. Section 2 analyzes related works in the area of cross-language filtering and retrieval. An overview of the approaches is provided in Section 3. Experiments carried out in a movie recommendation scenario are described in Section 4. Conclusions and future work are drawn in the last section.

2 Related Work

The Multilingual Information Filtering task at CLEF 2009[1] has introduced the issues related to the cross-language representation in the area of Information Filtering. The use of distributional models [2] in the area of monolingual and multilingual Information Filtering is a relatively new topic. Recently the research about semantic vector space models gained more and more attention: Semantic Vectors (SV)[2] package implements a Random Indexing algorithm and defines a negation operator based on quantum logic. Some initial investigations about the effectiveness of the SV for retrieval and filtering tasks is reported in [3].

3 Description of the approaches

Learning profiles through MultiWordnet. In this approach we can imagine a general architecture composed by three main components: the *Content Analyzer* allows to obtain a language-independent document representation by using a Word Sense Disambiguation algorithm based on MultiWordnet [4]. Similary to WordNet, the basic building block of MultiWordNet is the synset (SYNonym SET), a structure containing sets of words with synonymous meanings, which represents a specific meaning of a word. In MultiWordNet, for example the Italian WordNet is aligned with the English one, so by processing textual descriptions of items in both the languages, a language-independent representation in terms of MultiWordNet synsets is obtained. The generation of the cross-language user profile is performed by the *Profile Learner*, using a naïve Bayes text classifier, since each document has to be classified as interesting or not with respect to the user preferences. Finally the *Recommender* exploits the cross-language user profiles to suggest relevant items by matching concepts contained in the semantic profile against those contained in the disambiguated documents.
Distributional Models. The second strategy used to represent items content in a semantic space relies on the distributional approach. This approach represents documents as vectors in a high dimensional space, such as `WordSpace` [2]. The core idea behind `WordSpace` is that words and concepts (and documents, as well) are represented by points in a mathematical space, and this representation is learned from text in such a way that concepts with similar or related meanings

[1] http://www.clef-campaign.org/2009.html
[2] http://code.google.com/p/semanticvectors/

are near to one another in that space (geometric metaphor of meaning). Therefore, semantic similarity between documents can be represented as proximity in a n-dimensional space. Since these techniques are expected to efficiently handle high dimensional vectors, a common choice is to adopt *dimensionality reduction* that allows for representing high-dimensional data in a lower-dimensional space without losing information. *Random Indexing* (RI) [2] targets the problem of dimensionality reduction by removing the need for the matrix decomposition or factorization since it is based on the concept of Random Projection: the idea is that high dimensional vectors randomly chosen are "nearly orthogonal". This yields a result that is comparable to orthogonalization methods, but saving computational resources. Given two corpus (one for language *L1* and another one for *L2*) we build two monolingual spaces S_{L1} and S_{L2} that share the same *random base* by following the procedure introduced in [3]. Since both spaces share the same random base it is possible to compare elements belonging to different spaces: for example we can compute how a user profile in S_{L1} is similar to an item in S_{L2} (or viceversa). This property is used to provide recommendations.

4 Experimental evaluation

The goal of the experimental evaluation was to measure the predictive accuracy of both the content-based multilingual recommendation approaches. We compared the language-independent user profiles represented through MultiWordNet sysnsets and the approaches based on distributional hypothesis (W-SV) and Random Indexing (W-RI), already presented in [3].

The experimental work has been performed on a subset of the MovieLens dataset[3] containing 40,717 ratings provided by 613 different users on 520 movies. The content information for each movie was crawled from both the English and Italian version of Wikipedia. User profiles are learned by analyzing the ratings stored in the MovieLens dataset while the effectiveness of the recommendation approaches has been evaluated by means of *Precision@n* ($n = 5, 10$). We designed four different experiments: In Exp#1 and Exp#2 we learned user profiles on movies with English (respectively, Italian) description and recommended movies with Italian (respectively English) description and we compared their accuracy with the classical monolingual baselines calculated in Exp#3 and Exp#4. Results of the experiments are reported in Table 1, averaged over all the users.

In general, the main outcome of the experimental session is that the strategy implemented for providing cross-language recommendations is quite effective for both the approaches. Specifically, the approach based on the bayesian classifier gained the best results in the *Precision@5*. This means that model has a higher capacity to rank the best items at the top of the recommendation list. On the other side, the absence of a linguistic pre-processing is one of the strongest point of the approaches based on the distributional model and the results gained by the W-SV and W-RI models in the *Precision@10* further underlined the effectiveness of this model. In conclusion, both the approaches gained good results. Even

[3] http://www.grouplens.org

Table 1. Precision@5 and Precision@10

	Precision@5			Precision@10		
Experiment	**W-SV**	**W-RI**	**Bayes**	**W-SV**	**W-RI**	**Bayes**
EXP#1 – ENG-ITA	84,65	84,65	85,61	84,73	84,43	84,60
EXP#2 – ITA-ENG	84,85	84,63	85,20	84,77	84,54	84,56
EXP#3 – ENG-ENG	85,23	85,29	85,23	85,10	84,86	84,89
EXP#4 – ITA-ITA	85,27	84,84	85,71	85,11	84,86	84,93

though in most of the experiments the cross-lingua recommendation approaches get worse results w.r.t. the mono-lingual ones, the difference in the predictive accuracy does not appear statistically significant. In general the bayesian approach fits better in scenarios where the number of items to be represented is not too high, and this can justify the application of the pre-processing steps required for building the MultiWordNet synset representation, while the distributional models, thanks to their simplicity and effectiveness, fit better in scenarios where real-time recommendations need to be provided.

5 Conclusions

This paper compared two approaches for providing cross-language recommendations. The key idea is to provide a bridge among different languages by exploiting a language-independent representation of documents and user profiles based on word meanings. Experiments were carried out in a movie recommendation scenario, and the main outcome is that the accuracy of cross-language recommmendations is comparable to that of classical (monolingual) content-based recommendations.

References

1. C. Musto, F. Narducci, P. Basile, P. Lops, M. de Gemmis, and G. Semeraro, "Cross-language information filtering: Word sense disambiguation vs. distributional models," in *AI*IA*, 2011, pp. 250–261.
2. M. Sahlgren, "The Word-Space Model: Using distributional analysis to represent syntagmatic and paradigmatic relations between words in high-dimensional vector spaces," Ph.D. dissertation, Stockholm University, Department of Linguistics, 2006.
3. C. Musto, "Enhanced vector space models for content-based recommender systems," in *Proceedings of the fourth ACM conference on Recommender systems*, ser. RecSys '10. New York, NY, USA: ACM, 2010, pp. 361–364.
4. E. Pianta, L. Bentivogli, and C. Girardi, "MultiwordNet: developing an aligned multilingual database," in *Proc. of the 1st Int. WordNet Conference, Mysore, India*, 2002, pp. 293–302.

Using Snippets in Text Summarization: a Comparative Study and an Application

Giuliano Armano, Alessandro Giuliani, and Eloisa Vargiu

Abstract Automatic text summarization consists of automatically creating a summary of one or more texts. As for Web pages, unfortunately classical techniques cannot be applied in presence of dynamic contents. In this paper, we propose the adoption of snippets –i.e., page excerpts provided together with user query results by search engines– as a text summarization technique. The study is conducted along two directions: comparing the proposed approach with a classical text summarization technique and (ii) assessing whether snippet summarization can be successfully applied to contextual advertising. On the one hand, comparative experiments show that the proposed approach has performances similar to those obtained by using the selected classical technique. On the other hand, the adoption of snippets as text summarization technique in contextual advertising show that the performances are quite satisfactory.

1 Introduction

During the 60's, a large amount of scientific papers and books have been digitally stored and made searchable. Due to the limitation of storage capacity, documents were stored, indexed, and made searchable only through their summaries [29]. For this reason, how to automatically create summaries became a primary task and several techniques were defined and developed [18, 12, 25].

More recently, there has been a renewed interest on automatic summarization techniques. The problem now is no longer due to limited storage capacity, but to retrieval and filtering needs. Since digitally stored information is more and more available, users need suitable tools able to select, filter, and extract only relevant information. Therefore, text summarization techniques are currently adopted in sev-

G. Armano, A. Giuliani, and E. Vargiu
University of Cagliari, Dept.of Electrical and Electronic Engineering, Piazza d'Armi, I09123 Cagliari (Italy) e-mail: {armano, alessandro.giuliani, vargiu}@diee.unica.it

eral fields of information retrieval and filtering [7], such as, information extraction [21], text mining [31], document classification [27], recommender systems [23], and contextual advertising [1].

Unfortunately, classical techniques are not easily applicable to dynamic Web pages, which often rely on Microsoft Silverligh[1], Adobe Flash[2], Adobe Shockwave[3], or contain applets written in Java. Conventional parsing methods are often not applicable for the created webpage. Therefore, we claim that snippets, which are provided together with user query results by search engines, might be adopted to perform text summarization on Web pages.

In this paper, we are interested in studying the impact of snippets to perform text summarization. In particular, we conduct the study along two directions: (i) comparing performances obtained by using snippets with those obtained by adopting one of the classical text summarization techniques proposed in [3] and (ii) adopting snippets as text summarization technique in a selected application field, i.e., contextual advertising.

The rest of the paper is organized as follows. Section 2 recalls the main work on text summarization and introduces snippets and their use in search engines. Section 3 presents comparative experiments obtained by adopting snippets with respect to a classical text summarization technique. In Section 4, an application of snippet text summarization in the field of contextual advertising is proposed. Section 5 ends the paper with conclusions and future work.

2 Background

2.1 Text Summarization

Automatic text summarization is a technique in which a text is summarized by a computer program. Given a text, its summary (i.e., a non redundant extract from the original text) is returned.

Mani [19] made a distinction among different kinds of summaries: an *extract* consists entirely of material copied from the input; an *abstract* contains material that is not present in the input or, at least, expresses it in a different way; an *indicative abstract* is aimed at providing a basis for selecting documents for closer study of the full text; an *informative abstract* covers the salient information in the source at some level of detail; and a *critical abstract* evaluates the subject matter of the source document, expressing the abstractor views on the quality of the author's work.

According to [15], summarization techniques can be divided in two groups: those that extract information from the source documents (*extraction-based approaches*) and those that abstract from the source documents (*abstraction-based approaches*).

[1] http://www.microsoft.com/silverlight/

[2] http://www.adobe.com/products/flashplayer.html

[3] http://get.adobe.com/it/shockwave/

The former impose the constraint that a summary uses only components extracted from the source document. These approaches put strong emphasis on the form, aiming to produce a grammatical summary, which usually requires advanced language generation techniques. The latter relax the constraints on how the summary is created. These approaches are mainly concerned with what the summary content should be, usually relying solely on extraction of sentences.

Although potentially more powerful, abstraction-based approaches have been far less popular than their extraction-based counterparts, mainly because generating the latter is easier. An extraction-based summary consists of a subset of words from the original document and its bag of words (BoW) representation can be created by selectively removing a number of features from the original term set. Typically, an extraction-based summary whose length is only 10-15% of the original is likely to lead to a significant feature reduction as well. Many studies suggest that also simple summaries are quite effective in carrying over the relevant information about a document. Straightforward but effective extraction-based text summarization techniques have been proposed and compared in [15]. In a subsequent work, Armano et al. [3] proposed some enriched techniques. In particular, they showed that the technique with best performances in terms of precision, recall, and $F_{measure}$ was the so-called $TFLP$, i.e., the technique that considers the title of the document and its first and last paragraphs.

One may argue that extraction-based approaches are too simple. However, as shown in [9], extraction-based summaries of news articles can be more informative than those resulting from more complex approaches. Also, headline-based article descriptors proved to be effective in determining user's interests [14]. Moreover, these approaches have been successfully applied in the contextual advertising field [5] and in a multimodal scenario [2].

Fig. 1 An example of results given by Yahoo! search engine for the query "Information retrieval".

2.2 Snippets in Search Engines

A general definition of *snippet* is "a small piece of something". In programming, it refers to a small region of reusable source code, machine code, or text. Snippets are often used to clarify the meaning of an otherwise cluttered function, or to minimize the use of repeated code that is common to other functions.

Snippets are also used by search engines to provide a textual excerpt of the corresponding Web page according to the keywords used in the query. Snippet can be considered as a topic-driven summarization, since the summary content depends on the preferences of the user and can be assessed via a query, making the final summary focused on a particular topic. In a preliminary work, Boydell used snippets as summary fragments in the field of social Web [8].

While replying to a user's query, search engines provide a ranked list of related Web pages, each described by a title, a set of snippets, and its URL (see Figure 1). The title is directly taken from the *title* tag of the page, whereas the URL is the *http* address of the page.

For a search engine, the choice of a snippet is an important task. If a snippet shown to the user is not very informative, the user may click on search results that do not contain the information s/he is looking for, or s/he may not click on helpful pages. Moreover, poorly chosen snippets can lead to bad searching experiences. Snippets are usually directly taken from the *description meta* tag, if available. If the description meta tag is not provided, the search engine may use the description for the site supplied by the Open Directory Project (aka, DMoz)[4] or a summary extracted from the main content of the page.

Snippet extraction depends on the adopted search engine. Google[5] does not always use the meta description of the page. In fact, if the content provided by the Web developer in the description meta tag is not helpful, or less than reasonable quality, then Google replaces it with its own description of the site. In so doing, Google snippets will be different, depending on the user's search query. Yahoo![6] provides a patent application that describes how to better decide which snippet to show to users. The gist of Yahoo! patent application is based on three main issues[7]: (i) a query-independent relevance for each line of text, i.e., a degree to which the line of text of the document summarizes the document; (ii) a query-dependent relevance of each of the lines of text, i.e., a relevance of the line of text to the query; and (iii) the intent behind a query. To our best knowledge, Bing[8] developers do not give information on how snippets are extracted. In the literature there are several studies focused on the techniques of snippet extraction, usually relying on algorithms of natural language processing, e.g., as proposed by Li [17].

[4] http://dmoz.org

[5] http://www.google.com

[6] http://www.yahoo.com

[7] http://www.seobythesea.com/2009/12/how-a-search-engine-may-choose-search-snippets/

[8] http://www.bing.com

3 Comparative Study and Results

The first goal of this paper is to compare performances obtained by using snippets with those obtained by adopting a classical text summarization technique. Comparative experiments and the corresponding results are presented in this Section.

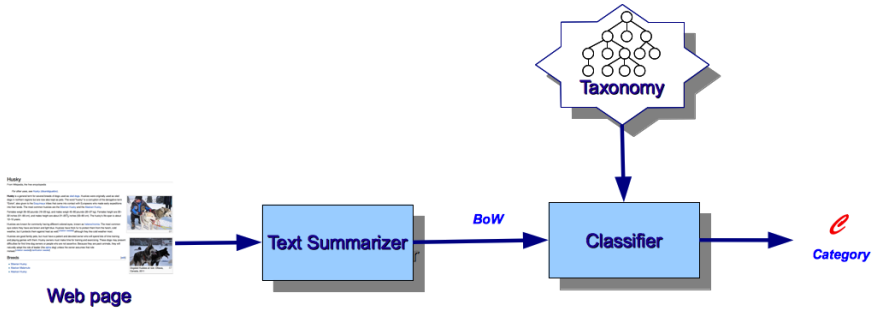


Fig. 2 The system adopted to perform comparative experiments on text summarization.

To perform comparative experiments, we devised a suitable system, depicted in Figure 2, in which the *Text Summarizer* module performs text summarization and the *Classifier* module is a centroid-based classifier aimed at classifying each page in order to calculate precision, recall and $F_{measure}$ of the adopted text summarization techniques. In other words, to assess the text summarization techniques, we used a Rocchio classifier [24] with only positive examples and no relevance feedback, preliminary trained with about 100 Web pages for class. Pages are classified by considering the highest score(s) obtained by the cosine similarity method. To evaluate the effectiveness of the classifier, we performed also a preliminary experiment in which pages are classified without relying on text summarization. The classifier showed a precision of 0.862 and a recall of 0.858.

3.1 Setting Up the Experiments

Experiments have been performed on two datasets extracted by the Open Directory Project and Yahoo! Categories. The former, called BankSearch [28], consists of about 11000 Web pages classified by hand in 11 categories (see Figure 3)[9]. The latter, called Recreation, consists of about 5000 Web pages classified by hand in 18 categories (see Figure 4).

[9] The 11 selected classes are the leaves of the taxonomy, together with the class *Sport*, which contains Web documents from all the sites that were classified as *Sport*, except for the sites that were classified as *Soccer* or *Motor Sport*.

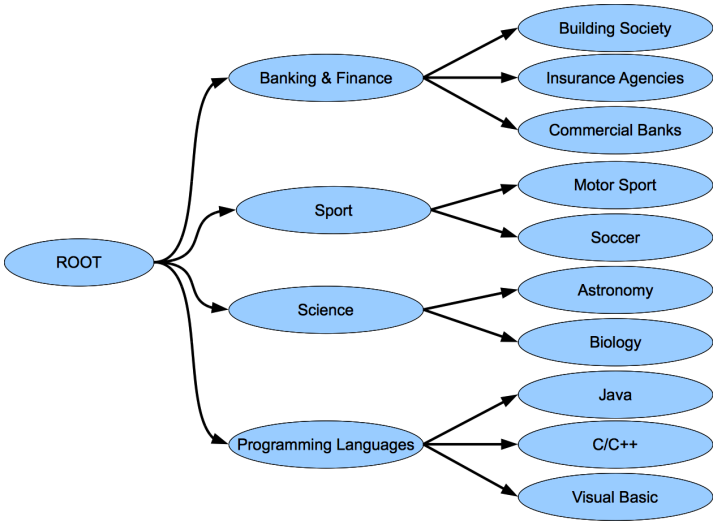


Fig. 3 The taxonomy of BankSearch Dataset.

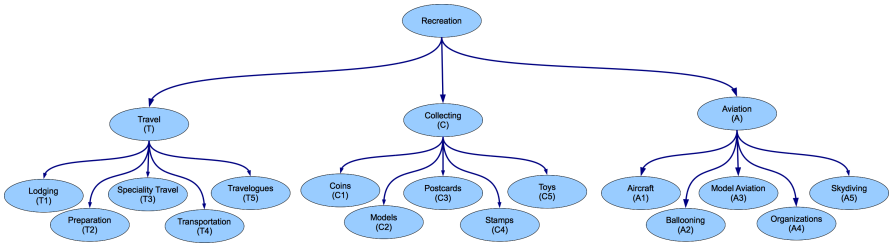


Fig. 4 The taxonomy of Recreation Dataset.

As a baseline for our comparative experiments, we adopted the text summarization technique called *TFLP* (Title, First and Last Paragraph summarization), which considers the title and the first and last paragraphs of the given Web page. This technique, proposed in [3], showed the best results compared with the state-of-the-art techniques proposed in [15]. As for snippets, we performed queries to Yahoo!, asking for the url of each webpage of the dataset, and we used the returned snippets. We performed experiments by considering the snippets by themselves (S) and in conjunction with the title of the corresponding Web page (ST). It is worth noting that we disregarded dynamic pages from both datasets in order to process the same number of pages independently by the adopted text summarization technique to perform a fair comparison.

3.2 Results

Table 1 reports our experimental results in terms of precision (π), recall (ρ), and $F_{measure}$ (F_1). The Table gives also the average number of extracted terms (T).

The results obtained on BankSearch are better than those obtained on Recreation. Moreover, they point out that, in both datasets, results obtained by relying on snippets together with the title (ST) are comparable with those obtained by adopting $TFLP$. In particular, $TFLP$ performs slightly better in BankSearch, whereas ST performs slightly better in Recreation. This proves that snippets can be adopted as text summarization techniques, especially when classical techniques can not be applied, as in the case of dynamic Web pages.

Let us note that, for each dataset, the average number of terms for the TFLP technique is about twice the number of terms for the method that uses to snippets. This is due to the fact that a snippet is built as a very short text, not less than two rows, wheres in a TFLP summary is usually longer (two complete paragraphs).

Table 1 Results of text summarization techniques comparison.

	BankSearch			**Recreation**		
	TFLP	**S**	**ST**	**TFLP**	**S**	**ST**
π	**0.849**	0,734	0.806	0.575	0.544	**0.595**
ρ	**0.845**	0.730	0.804	**0.556**	0.506	0.554
F_1	**0.847**	0.732	0.805	0.565	0.524	**0.574**
T	26	12	14	26	11	13

4 Using Snippets as Text Summarization Technique in Contextual Advertising

The second goal of this paper is to study the impact of snippet text summarization in a selected application field. Among other relevant information retrieval and filtering fields in which snippet text summarization could be adopted, we concentrate on contextual advertising.

4.1 Contextual Advertising

Web advertising is one of the major sources of income for a large number of websites. Its main goal is to suggest products and services to the ever growing population of Internet users. There are two primary channels for distributing ads: Sponsored Search (or Paid Search Advertising) and Contextual Advertising (or Content Match). Sponsored Search displays ads on the page returned from a search engine

following a query [13]; whereas Contextual Advertising (CA) displays ads within the content of a generic, third party, Web page.

Ribeiro-Neto et al. [22] examined a number of strategies to match pages and ads based on extracted keywords. In a subsequent work, Lacerda et al. [16] proposed a method to learn the impact of individual features using genetic programming. Broder et al. [10] classified both pages and ads into a given taxonomy and matched ads to the page falling into the same node of the taxonomy. Starting from that work, Armano et al. [4] proposed a semantic enrichment by adopting concepts. Furthermore, modern contextual advertising systems use text summarization techniques in conjunction with the model developed in [10] (see, for instance [1, 5]). Since bid phrases are basically search queries, another relevant approach is to view contextual advertising as a problem of query expansion and rewriting [20, 11]. Another perspective consists on addressing a contextual advertising problem as a recommendation task [6]. Thus, authors view the task of suggesting an ad to a Web page as the task of recommending an item (the ad) to a user (the Web page).

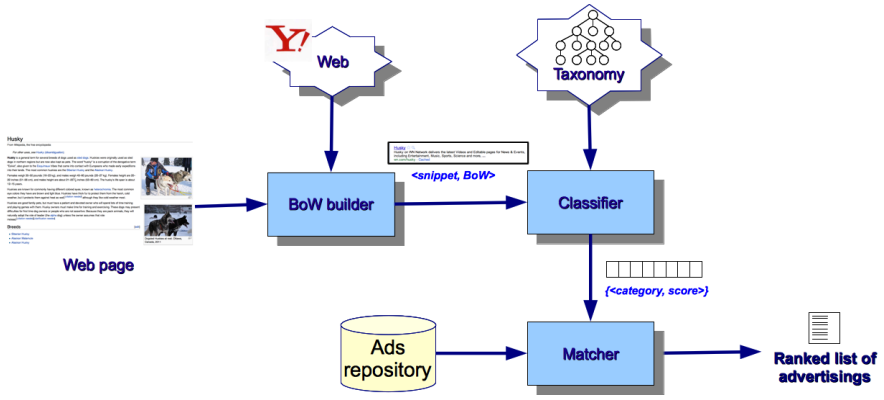


Fig. 5 The implemented contextual advertising system.

4.2 The Implemented System

Being interested in studying the impact of snippets as text summarization technique in contextual advertising, we devised a suitable system (see Figure 5). The system takes a Web page as input. The *BoW builder*, first, retrieves the snippets of the page by asking to Yahoo! search engine and then removes stop-words and performs stemming. This module outputs a vector representation of the original text as *BoW*, each word being represented by its TFIDF [26]. Starting from the *BoW* provided by the *BoW builder*, the *Classifier* classifies the page according to the given taxonomy by adopting a centroid-based approach. This module outputs a vector representation

in terms of Classification Features (CF), each features corresponding to the score given by the classifier to each category. Finally, the *Matcher* ranks the categories according to the scores given by the classifier (i.e., the CF of the target page) and, for each category, randomly extracts a corresponding ad from the *Ads repository*.

Let us note that the proposed system, except for the adopted text summarization technique, is compliant with the system proposed in [1] in which only CF are considered in the matching phase.

4.3 System Performances

To assess the effectiveness of the proposed approach, experiments have been performed on the Recreation dataset described in Section 3.1. As for the ads to be suggested, we built a suitable repository in which ads are classified according to the given taxonomy. In this repository, each ad is represented by the Web page of a product or service company.

Performances have been calculated in terms of *precision at k* with $k \in [1,5]$, i.e., the precision in suggesting k ads. Given a page p and an ad a, the $\langle p,\ a\rangle$ pair has been scored on a 1 to 3 scale defined as follows:

1 - Relevant: a is semantically directly related to the main subject of p, i.e., a and p belongs to the same category;
2 - Somewhat relevant: (i) a is related to a similar subject of p (*sibling*), i.e., a and p belongs to sibling categories; (ii) a is related to the main topic of p in a more general way (*generalization*), i.e., a belongs to the parent node of the category p; or (iii) a is related to the main topic of p in a too specific way (*specification*), i.e., a belongs to a child of the category of p;
3 - Irrelevant. a is unrelated to p, i.e., the category to which a belongs is in a different branch with respect to the category to which p belongs.

According to state-of-the-art contextual advertising systems (e.g., [10]), we considered as True Positives (TP) ads scored as 1 or 2, and a False Positives (FP) ads scored as 3.

Table 2 Precision at k of the proposed contextual advertising system by adopting: $TFLP$ (CA_{TFLP}), the sole snippets (CA_S); and the snippets together with the page title (CA_{ST}).

k	$\mathbf{CA}_{TFLP}$	$\mathbf{CA}_S$	$\mathbf{CA}_{ST}$
1	**0.868**	0.837	0.866
2	0.835	0.801	**0.836**
3	0.770	0.746	**0.775**
4	0.722	0.701	**0.729**
5	0.674	0.657	**0.681**

In performing experiments, we compared the performances obtained by using as text summarization technique: $TFLP$, the resulting system being CA_{TFLP}; the

sole snippets, the resulting system being CA_S; and the snippets together with the page title, the resulting system being CA_{ST}. Let us note that, as the focus of this paper is on text summarization, comparative experiments among the implemented contextual advertising system and selected state-of-the-art systems are out of the scope of this work. Nevertheless, let us stress that CA_{TFLP} coincides with the system proposed in [5] in which the α parameter is set to 0 (i.e., only CF are considered in the matching phase).

Table 2 shows that, for all the compared systems, results are quite satisfactory, especially in suggesting 1 or 2 ads. It also clearly shows that, except for $k = 1$, CA_{ST} is the system that performs better. This proves the effectiveness of adopting snippets as text summarization technique in the field of contextual advertising.

5 Conclusions and Future Work

Since classical text summarization techniques are not applicable for dynamic Web pages, in this paper we proposed to use snippets. The aim of the paper was twofold: (i) to compare performances obtained by using snippets with those obtained by adopting a classical text summarization technique and (ii) to study the impact of snippets in a selected application field, i.e., contextual advertising. The comparisons showed that the proposed snippet text summarization technique has performances (in terms of precision, recall, and F_1) similar to those obtained by using a classical technique (i.e., $TFLP$). The adoption of snippets as text summarization technique in contextual advertising showed that performances, calculated in terms of precision at k, are quite good, especially in suggesting 1 or 2 ads, and that the system that uses both snippets and title is the one with the best performances.

As for future work we are planning to perform further comparative experiments with the methods described in [18, 30, 12].

Acknowledgment

This work has been partially supported by Hoplo srl. We wish to thank, in particular, Ferdinando Licheri and Roberto Murgia for their help and useful suggestions.

References

1. Anagnostopoulos, A., Broder, A.Z., Gabrilovich, E., Josifovski, V., Riedel, L.: Just-in-time contextual advertising. In: CIKM '07: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management, pp. 331–340. ACM, New York, NY, USA (2007). DOI http://doi.acm.org/10.1145/1321440.1321488

2. Armano, G., Giuliani, A., Messina, A., Montagnuolo, M., Vargiu, E.: Experimenting text summarization on multimodal aggregation. In: 5th International Workshop DART 2011, New Challenges on Information Retrieval and Filtering, CEUR Workshop Proceedings, Vol. 771. C. Lai and G. Semeraro and E. Vargiu (2011)
3. Armano, G., Giuliani, A., Vargiu, E.: Experimenting text summarization techniques for contextual advertising. In: IIR'11: Proceedings of the 2nd Italian Information Retrieval (IIR) Workshop (2011)
4. Armano, G., Giuliani, A., Vargiu, E.: Semantic enrichment of contextual advertising by using concepts. In: International Conference on Knowledge Discovery and Information Retrieval (2011)
5. Armano, G., Giuliani, A., Vargiu, E.: Studying the impact of text summarization on contextual advertising. In: 8th International Workshop on Text-based Information Retrieval (2011)
6. Armano, G., Vargiu, E.: A unifying view of contextual advertising and recommender systems. In: Proceedings of International Conference on Knowledge Discovery and Information Retrieval (KDIR 2010), pp. 463–466 (2010)
7. Baeza-Yates, R.A., Ribeiro-Neto, B.: Modern Information Retrieval. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA (1999)
8. Boydell, O., Smyth, B.: From social bookmarking to social summarization: an experiment in community-based summary generation. In: Proceedings of the 12th international conference on Intelligent user interfaces, IUI '07, pp. 42–51. ACM, New York, NY, USA (2007)
9. Brandow, R., Mitze, K., Rau, L.F.: Automatic condensation of electronic publications by sentence selection. Inf. Process. Manage. **31**, 675–685 (1995)
10. Broder, A., Fontoura, M., Josifovski, V., Riedel, L.: A semantic approach to contextual advertising. In: SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 559–566. ACM, New York, NY, USA (2007). DOI http://doi.acm.org/10.1145/1277741.1277837
11. Ciaramita, M., Murdock, V., Plachouras, V.: Online learning from click data for sponsored search. In: Proceeding of the 17th international conference on World Wide Web, WWW '08, pp. 227–236. ACM, New York, NY, USA (2008)
12. Edmundson, H.P.: New methods in automatic extracting. J. ACM **16**, 264–285 (1969)
13. Feldman, J., Muthukrishnan, S.: Algorithmic methods for sponsored search advertising. CoRR **abs/0805.1759** (2008)
14. Kołcz, A., Alspector, J.: Asymmetric missing-data problems: Overcoming the lack of negative data in preference ranking. Inf. Retr. **5**, 5–40 (2002)
15. Kolcz, A., Prabakarmurthi, V., Kalita, J.: Summarization as feature selection for text categorization. In: CIKM '01: Proceedings of the tenth international conference on Information and knowledge management, pp. 365–370. ACM, New York, NY, USA (2001)
16. Lacerda, A., Cristo, M., Gonçalves, M.A., Fan, W., Ziviani, N., Ribeiro-Neto, B.: Learning to advertise. In: SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 549–556. ACM, New York, NY, USA (2006). DOI http://doi.acm.org/10.1145/1148170.1148265
17. Li, Q., Chen, Y.P.: Personalized text snippet extraction using statistical language models. Pattern Recogn. **43**, 378–386 (2010)
18. Luhn, H.P.: The automatic creation of literature abstracts. IBM Journal of Research and Development **2**(2), 159–165 (1958)
19. Mani, I.: Automatic summarization. John Benjamins, Amsterdam (2001)
20. Murdock, V., Ciaramita, M., Plachouras, V.: A noisy-channel approach to contextual advertising. In: Proceedings of the 1st international workshop on Data mining and audience intelligence for advertising, ADKDD '07, pp. 21–27. ACM, New York, NY, USA (2007)
21. Rau, L.F., Jacobs, P.S., Zernik, U.: Information extraction and text summarization using linguistic knowledge acquisition. Inf. Process. Manage. **25**, 419–428 (1989)
22. Ribeiro-Neto, B., Cristo, M., Golgher, P.B., Silva de Moura, E.: Impedance coupling in content-targeted advertising. In: SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 496–503. ACM, New York, NY, USA (2005). DOI http://doi.acm.org/10.1145/1076034.1076119

23. Ricci, F., Rokach, L., Shapira, B., Kantor, P.: Recommender Systems Handbook. Springer, US (2010)
24. Rocchio, J.: The SMART Retrieval System: Experiments in Automatic Document Processing, chap. Relevance feedback in information retrieval, pp. 313–323. PrenticeHall (1971)
25. Salton, G., Buckley, C.: On the use of spreading activation methods in automatic information. In: Proceedings of the 11th annual international ACM SIGIR conference on Research and development in information retrieval, SIGIR '88, pp. 147–160. ACM, New York, NY, USA (1988)
26. Salton, G., McGill, M.: Introduction to Modern Information Retrieval. McGraw-Hill Book Company (1984)
27. Shen, D., Chen, Z., Yang, Q., Zeng, H.J., Zhang, B., Lu, Y., Ma, W.Y.: Web-page classification through summarization. In: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval, SIGIR '04, pp. 242–249. ACM, New York, NY, USA (2004)
28. Sinka, M., Corne, D.: A large benchmark dataset for web document clustering. In: Soft Computing Systems: Design, Management and Applications, Volume 87 of Frontiers in Artificial Intelligence and Applications, pp. 881–890. Press (2002)
29. de Smedt, K., Liseth, A., Hassel, M., Dalianis, H.: How short is good? An evaluation of automatic summarization, pp. 267–287. Museum Tusculanums Forlag, Kbenhavn (2005)
30. Tsegay, Y., Puglisi, S.J., Turpin, A., Zobel, J.: Document compaction for efficient query biased snippet generation. In: Proceedings of the 31th European Conference on IR Research on Advances in Information Retrieval, ECIR '09, pp. 509–520. Springer-Verlag, Berlin, Heidelberg (2009)
31. Witten, I.H., Bray, Z., Mahoui, M., Teahan, B.: Text mining: A new frontier for lossless compression. In: Proceedings of the Conference on Data Compression, DCC '99, pp. 198–. IEEE Computer Society, Washington, DC, USA (1999)

Grammatical Feature Engineering for fine-grained IR tasks

Danilo Croce and Roberto Basili

Department of Enterprise Engineering
University of Roma, Tor Vergata
{croce,basili}@info.uniroma2.it

Abstract. Information Retrieval tasks include nowadays more and more complex information in order to face contemporary challenges such as Opinion Mining (OM) or Question Answering (QA). These are examples of tasks where complex linguistic information is required for reasonable performances on realistic data sets. As natural language learning is usually applied to these tasks, rich structures, such as parse trees, are critical as they require complex resources and accurate pre-processing. In this paper, we show how good quality language learning methods can be applied to the above tasks by using grammatical representations simpler than parse trees. These features are here shown to achieve the state-of-art accuracy in different IR tasks, such as OM and QA.

1 Syntactic modeling of linguistic features in Semantic Tasks

Information Retrieval faces nowadays contemporary challenges such as Sentiment Analysis (SA) or Question Answering (QA), that are tight to complex and fine grained linguistic information. The traditional view in IR that represents the meaning of documents just according to the words that occur in them is not directly applicable. Statistical models, such as the vector-space model or variants of the probabilistic model that express documents and queries as Bags-of-Words (BOW) [1] are too poor. Even though fully lexicalized models are well established, in recent years syntactic and semantic structures expressing richer linguistic structures are becoming essential in complex IR tasks, such as Question Classification [21] and Passage Ranking [3] in Question Answering (QA) or Sentiment Analysis Opinion Mining (OM) [12]. The major problem here is that fine-grained phenomena are targeted, and lexical information alone is not sufficient.

The capabilities of the BOW retrieval models do not alway provide a robust solution to these real retrieval needs. For example, in a QA system a BOW IR retrieves documents matching a query, but the QA system actually needs documents that contain answers. The question analysis is thus crucial for the QA system to model the user information needs and to retrieve a proper answer. This is made available when the linguistic and semantic constraints imposed by the question are satisfied by an answer, thus requiring a effective selection of answer-bearing passages.

Language learning systems allow to generalize linguistic observations into rules and patterns as statistical models of higher level semantic inferences. Statistical learning methods make the assumption that lexical or grammatical observations are useful

hints for modeling different semantic inferences, such as in document topical classification, predicate and role recognition in sentences as well as question classification in Question Answering. Lexical features here include lemmas, multiword expressions or Named Entities that can be directly observed in the texts. Features are then generalized into predictive components in the final model, induced from the training examples. Obviously, lexical information usually implies different words to provide different contributions but usually neglect other crucial linguistic properties, such as word ordering.

The information about the sentence syntactic structure can be thus exploited and symbolic expressions derived from the parse trees of training examples are used as features for language learning systems. These features denote the position and the relationship between words that can be seemingly realized by different trees independently from irrelevant differences. For example, in a declarative sentence (such as in a S←NP VP structure), the relationship between a verbal predicate (VP) and its immediately preceding grammatical subject (NP) is literally translated in the feature VP↑VP↑S↓NP, where arrows indicate upward or downward movements through the tree. Linear kernels over the resulting *Parse Tree Path* features are employed in NLP tasks such as for Semantic Role Labeling [14] or Opinion Mining [22]. This idea is further expanded in tree kernels, introduced by [5]. These model similarity between training examples as a function of the shared subtrees in their corresponding parses. Tree kernels have been successfully applied to different tasks ranging from parsing [5] to semantic role labeling [19]. Tree kernels are known to determine a better grammatical representation for the targeted examples and provide an implicit method for robust feature engineering.

However, the adoption of grammatical features and tree kernels is still affected by significant drawbacks. First, strict requirements exist in terms of the size of the training data set as high dimensionality spaces are generated, whose data sparseness can be prohibitive. Usually, the application of exact learning algorithms gives rise to complex training processes whose convergence is quite slow. Although specific forms of optimization have been proposed to limit their inherent complexity (e.g. [18]), tree kernels do not scale well over very large training data sets. Finally it must be noticed that most of the methods extracting grammatical features from parse trees, are strongly biased by parsing errors.

We want to explore here a possible solution to the above problems through the adoption of shallow but more consistent grammatical features that avoid the use of a full parser in semantic tasks. Parsing accuracy is highly varying across corpora, and it is often poorly effective for some natural languages or application domains where limited resources are available or the syntactic structure of the test instances is very different with respect to the training material. In particular [7] investigates the accuracy loss of well known syntactic parsers applied to micro-blogging datasets. In particular they observed a drastic drop in performance moving from the in-domain test set to the new Twitter dataset. Avoiding the adoption of full parsing obviously increases the number and nature of possible uses of language technologies in a variety of complex NLP applications. In IR, part of speech information has been generally used for stemming, generating stop-word lists, and identifying pertinent terms or phrases in documents and/or in queries. Generally, the state of the art in IR systems tend to benefit from the adoption of parts of speech to index or retrieve information [24].

The open research questions are: which shallow grammatical representation is suitable to support the learning of fine-grained semantic models? Which grammatical generalizations can be usefully achieved over shallow syntactic representations for sentence-based inferences?

In the rest of this work, we show how embedding shallow grammatical information in a sentence representation, as a special case of enriched lexical information, produces useful generalizations in standard machine learning settings. Empirical findings in support to this thesis are discussed against two complex sentence-based semantic tasks, i.e. question classification and sentiment analysis in micro-blogging.

2 Shallow Parsing and Grammatical Feature engineering

Grammatical feature engineering is required as lexical information alone is, in general, not sufficient to characterize linguistic generalizations useful for fine-grained semantic inferences. For example, sentence (3) is the appropriate answer for the question (1), although both sentences (2) and (3) are reasonable candidates.

What ***French province is Cognac produced in****?* (1)

The grapes which ***produce*** *the* ***Cognac*** *grow* ***in*** *the* ***province*** *and the* ***French*** *government ...* (2)

Cognac is *a brandy* ***produced in*** *Poitou-Charentes.* (3)

Suppose we use a lexical overlap rule for a Question Answering (QA) task: given the overlapping terms outlined in bold[1], it would result in the wrong answer (2). A simple lexical overlap model is too simplistic, as syntactic information characterizing the individual sentences (1) and (3) is here necessary. Syntactic features provide more information to estimate the similarity between the question and the candidate answers, as in general explored by tree kernels in Answer Classification/Re-ranking [20]. The parse tree in Figure 1 corresponds to sentence (3) and represents:

- *lexical information* through its terminal nodes (e.g., words as $Cognac,\ is,\ \ldots$)
- *Coarse-grained grammatical information* through the POS tag characterizing pre-terminal nodes (e.g. NNP or VBZ)
- *Fine-grained grammatical information* as subtrees correspond to the production rules of the underlying *context free grammar* (CFG).

Examples of the CFG rules involved in Figure 1 are: $S \rightarrow NP\ VP$, $VP \rightarrow VBZ\ NP$, $NP \rightarrow NPP$ or $NP \rightarrow DT\ NN$. Stochastic context free grammars (e.g. [4]), are generative models for parse trees, seen as complex joint events, whose overall probability depends on the individual CFG rules (i.e., subtrees), and lexical information as well. Our aim here is to acquire these rules implicitly, as a side effect of the learning for semantic inference process. Specific features can in fact be designed to surrogate the syntactic structures of the parse tree, implicitly. Observable POS tag sequences correspond to subtrees and can be considered their shallow counterpart.

[1] Sentence (2) shares five terms with the sentence (1), while (3) shares only four terms.

They express linearly special properties, in analogy with the Parse Tree Paths in [9]. In other words, subtrees can be artificially replaced introducing POS tag sequences (or POS n-grams), instead of parse tree fragments. The idea is that the syntactic structure of a sentence could be surrogated as the POS n-grams, instead of the set of possible syntactic tree fragments, as used by tree kernels. For example, the partial tree expressed by VP→VBN PP in Fig. 1 can be represented through the pseudo token given by *VBN-IN-NNP*.

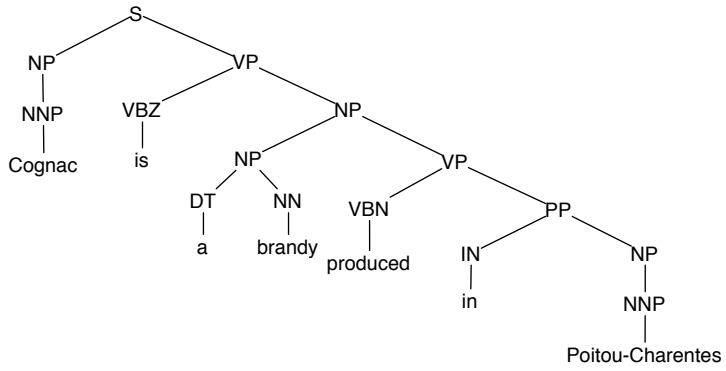


Fig. 1. Example of parse tree associated to sentence (3)

Lexicalized features (i.e., true words) as well as shallow syntactic information (i.e., the POS n-grams) are thus made available as flat features, thus constraining the capacity of the underlying learning machine. A sentence s of length $|s|$ is thus represented as a set of words (in a bag-of-word fashion), extended by the pseudo tokens defining the corresponding POS tag sequences whose length is smaller that n (n-POS tag grams). Given the word sequence $s = \{w_1, \ldots, w_{|s|}\}$ whose corresponding part-of-speeches are $\{pos_1, \ldots, pos_{|s|}\}$, the representation of the pseudo tokens is the set of pairs $\{(w_1.pos_1), \ldots, (w_{|s|}.pos_{|s|}\}$, where each lemmatized word is coupled with its POS tag.

Moreover, in order to capture syntactic structures of interest, POS tags are also mapped into pseudo-tokens expressing their sequences (i.e., POS n-grams). Given n as the maximal size of the extracted sequences, every subsequence of length at most n is mapped into a pseudo-token. These novel *grammatical* tokens of length Δ are expressed as $\{p_j, \ldots, p_{j+\Delta}\}$ where $\Delta = 1, ..., n$. In these patterns the representation of prepositions (POS tag *IN*) is made explicit. Every position $k \in [j, j + \Delta]$ for which pos_k=*IN* is represented through w_k itself, so that *at-NP* or *of-DT-NN* are obtained as pseudo-tokens for fragments such as "*at Whitlock*" or "*of the vineyard*". The representation of sentence (3) is shown in Table 2, where words $(w_i.pos_i)$ and n-gram tokens are shown.

Table 1. Representation of lexical and grammatical information for sentence (3)

unigrams	cognac.NNP be.VBZ a.DT brandy.NN produce.VBN in.IN poitou-charentes.NNP
2-grams	NNP-VBZ VBZ-DT DT-NN NN-VBN VBN-in in-NNP NNP-.
3-grams	NNP-VBZ-DT VBZ-DT-NN DT-NN-VBN NN-VBN-in VBN-in-NNP in-NNP-.
4-grams	NNP-VBZ-DT-NN VBZ-DT-NN-VBN DT-NN-VBN-in NN-VBN-in-NNP VBN-in-NNP-.

2.1 Shallow Syntactic Features for Question Classification

In Question Answering three main processing stages are foreseen: question processing, document retrieval and answer extraction [16]. Question processing is usually centered around the so called *question classification* (QC) task that maps a question into one of k predefined answer classes [17]. Typical examples of classes characterize different answer strategies and range from questions regarding *persons* or *organizations* (e.g. *Who killed JFK*?) to *definition* questions (e.g. *What is a perceptron*?) or *modalities* (e.g. *How fast does boiling water cool?*). Highly accurate QC systems apply supervised machine learning techniques, e.g. Support Vector Machines (SVMs) [20, 23] or the SNoW model [17], where questions are encoded using a variety of lexical, syntactic and semantic features. In [17], it has been shown that the questions' syntactic structure contributes remarkably to the classification accuracy. This task is thus strictly syntax-dependent, especially because individual sentences are targeted.

As questions can be regarded as individual sentences, we will adopt the feature extraction scheme proposed in Table 2 for our QC models. These features represent both lexical and grammatical information that can be efficiently feed a statistical classifier based on linear kernels. Section 3.1 will discuss comparative experiments with previous works on Question Classification.

2.2 Shallow Syntactic Features for Sentiment Analysis over micro-blogging

Microblogging has been already established as a significant form of electronic word-of-mouth for sharing opinions, suggestions and consumer reviews concerning ideas, products or brands. Microblogging is also referred to as micro-sharing or *Twittering* (from Twitter[2] by far the most popular microblogging application). While opinion mining over traditional text sources (e.g. movie reviews or forums) has been significantly studied [22], sentiment analysis over tweets has a more recent history, [10] or [2]. It has been usually addressed on the basis of only lexical information whereas the syntactic structure of tweets is often neglected [22]. In [25] the linguistic redundancy in Twitter is investigated and several types of linguistic features are tested in a supervised setting, showing that tweet syntactic structure does not provide alone a statistically significant contribution with respect to lexical typed features. The main problem of syntax-driven

[2] `http://www.twitter.com`

approaches over tweets is the quality of the available grammatical information as tweets are sentences lacking of a proper grammatical structure.

Here the modeling through POS n-grams is suitable to overcome these problems, as it provides a simpler representation of the tweets' syntax and, on the other hand, it should be more robust as for tagging accuracy. However even POS taggers, trained over standard texts, may be inadequate, as the linguistic form of tweets is rather non standard with a large use of jargon and shortcuts. An interesting finding in [7] was that one of the main cause of the syntactic parsing errors over the Twitter dataset is due to the propagation of part-of-speech tagging errors. In line with other works (see for example [10] or [15]), we propose to pre-process tweets before a *standard* POS tagger is applied. This avoids the noise in applying traditional POS tagging to odd symbols (e.g. re-tweets or emoticons) or jargon expressions and also reduces data sparseness, as canonical forms are adopted. The following set of actions is applied before training:

- fully capitalized words are first converted in their lowercase counterpart, i.e. "*DOG*" into "*dog*", before applying POS tagging
- reply marks (i.e. *@user_name*) are replaced with the pseudo-token `USER` whose POS tag is set back to `PUSER` after POS tagging
- hyperlinks are replaced by the token `LINK` whose POS is `PLINK`
- hash tags (i.e. *#thread_name*) are replaced by the pseudo-token `THREAD` whose POS is imposed to `PTHREAD`
- repeated letters and punctuation characters (e.g. *looove*, *loooove* or *!!!*) are cleansed as they cause high levels of lexical data sparseness. Characters occurring more than twice are all replaced with a double occurrence expression, so that *looove* or *!!!* are mapped into *loove* or *!!*, respectively
- all emoticons, e.g. :-) or *:P*, are used as sentence separators although they are systematically misinterpreted by a standard POS tagger. Accordingly, they are first replaced with a full stop "." and then recovered at their original form after POS tagging. Their POS is always set to `SMILE`.

After the above pre-processing phase, a tweet like *@jdoe I looove Twitter! :-) http://twitpic.com/2y2e0* can be represented according to the model proposed in Section 2. Here the lists of lexical unigrams and grammatical n-grams are reported:

```
USER.PUSER i.PRP loove.VBP twitter.NNP!.PUNC :-).SMILE LINK.PLINK
PUSER_PRP PRP_VBP VBP_NNP NNP_PUNC PUNC_SMILE SMILE_PLINK USER_PRP_VBP ...
```

As it is clear from the example, the resulting POS sequences are able to better capture the intended syntax and act as good models of relevant grammatical relations: the sequence `USER.PUSER i.PRP loove.VBP` ..., for example, is a good hint for the positive bias introduced by $loove$ as a verb.

3 Performance Evaluation

In this section we evaluate the use of POS n-grams in two applications previously discussed as standard example of different semantic inferences useful for IR. In all the

experiments POS tagging is carried out by the tagger available in the LTH parser [13]. The performance achievable by POS n-grams is thus compared with the one derived by richer grammatical representations based on parse trees.

3.1 Question Classification Results

This first experiment studies the impact of combining lexical and shallow syntactic information (i.e. POS n-grams), on question classification. The targeted dataset is the UIUC corpus, largely adopted for benchmarking [17]. UIUC contains a training set of 5,452 questions and a test set of 500 questions, both extracted from TREC. Question classes are organized in two levels of granularity. At the first level, 6 coarse-grained classes are defined, like ABBREVIATION, ENTITY, DESCRIPTION. A second level explodes the first level classes into a set of 50 fine-grained sub-classes, e.g., *Plant* and *Food* are subclasses of the ENTITY category.

SVM learning is applied over the feature vectors discussed in Section 2.1 and multi-classification is modeled through a *one-vs-all* scheme. The quality of classification is measured through accuracy, i.e. the percentage of questions associated with the correct class. A development set is derived from the 20% of the training material. In the experiments two sentence models are compared:

- *POS tagged Unigrams (PU)*: a question is mapped into a bag of POS tagged lemmas, i.e. into pairs of $(lemma.pos)$. This model is based only on lexical information.
- *POS n-grams (PnG)*: each question is modeled by augmenting the PU model through the shallow syntactic information provided by the sequence of n-grams of POS tags, with $n < 4$. The POS of Wh-determiners and prepositions are replaced in the individual POS n-grams by the corresponding lemmas.

In this evaluation the voted perceptron [8] and SMV^{light} [11] have been both applied[3] . Results, compared with the results achieved by the system discussed in [23] on the same UIUC dataset, are shown in Table 2. The authors combine a kernel classifier based on BOW with two semantic kernels: one (i.e. K(LS)) is based on Latent Semantic Indexing applied to Wikipedia, and the other (i.e. K(semRel)) uses semantic information acquired through *manually constructed lists of words*, i.e., a task-specific lexicon related to the answer types.

In the coarse-grained test, i.e. the question classification with respect to the 6 coarse grained classes, Table 2 shows how the syntactic generalization supported by the PnG model achieves the best known results on the UIUC dataset, i.e., 91.8% that correspond to the accuracy reported by a tree kernel approach [20], without any semantic extension. This improves the best results of [23] (i.e., the $K(BOW) + K(LS) + K(semRel)$) that refer to a *task-dependent use of manually annotated resources*. Note how the kernel $K(LS)$ that uses only lexical information, gathered by an external corpus like Wikipedia [23] is also weaker than the PnG model, that makes no use of trees or other

[3] In the experiments a polynomial kernel of degree 2 has been applied with SMV^{light}, as it achieved the best result on the development set

Table 2. Accuracy measures for the QC task

Kernel	Coarse Task	Fine-grain Task
PU (VotedPerc)	89.2%	81.4%
PU (SVM)	89.4%	83.8%
PnG (VotedPerc)	91.4%	84.0%
PnG (SVM)	**91.8%**	84.8%
[23]		
K(BOW)	86.4%	80.8%
K(LS)	70.4%	71.2%
K(BOW)+K(LS)	90.0%	83.2%
K(BOW)+K(LS)+K(semRel)	90.8%	**85.6%**
[20]		
Tree Kernels		
K(BOW)+K(*PartialTrees*)	**91.8%**	-

resources. The results in Table 2 are also remarkable from a computational point of view: the PnG method only requires POS tagged sentences and no parsing. Moreover, the training time of tree kernel based SVMs on benchmarking data sets are in the order of hours or days for large training collections (e.g., Prop Bank, as reported in [18]).

In [6] an extension to the tree kernel formulation has been proposed, i.e. the semantic Smoothed Partial Tree Kernel that enriches the similarity among syntactic tree structures with lexical information gathered by en external corpus, in line with the K(LS) described in [23]. State-of-the art results of 94.8% have been obtained in the coarse-grained test. However it is still a complex approach that need explicit syntactic parsing of the sentences and an external corpus that provides lexical knowledge. This is beyond the scope of this work, that aims at providing an efficient and practical engineering method for natural language learning systems. The training complexity of the proposed models is very low. Consider that for a short sentence (i.e. a question or a micro-blogging message) the number of feature is reduced. For example a sentence of 10 words, will generate 10 lexical, 9 bi-gram, 8 three-gram and 7 four-gram features, i.e. a feature vector of 34 features. It generates a hi-dimensional but very sparse space, where both SVM and the vote perceptron algorithms can very effectively find a solution. The efficiency of the proposed method in the QC task is thus proved, as the *PnG* model has been trained over 5,452 examples in less than 2 minutes and 40 seconds, with SMV^{light} and the voted perceptron, respectively.

3.2 Sentiment Analysis Results

The POS n-grams model has been also applied in the task of Sentiment Analysis over tweets, as introduced in Section 2.2. The goal here is to classify a tweet according to its sentiment polarity. The adopted dataset is Twitter Sentiment, released by [10][4], as other studies (e.g. [2]) do not allow a full comparative analysis. It provides a training

[4] `http://www.stanford.edu/~alecmgo/cs224n/twitterdata.2009.05.25.c.zip`

set automatically generated by selecting the positive (or negative) examples from the tweets containing positive (or negative) emoticons, e.g. :-) (or :-( ). The test set, also made available by [10], includes 183 tweets, manually annotated according to their binary sentiment polarity, i.e. ± 1. Each tweet is modeled as a feature vector, including words as well as the pseudo-tokens generated in the pre-processing phase, including the resulting POS n-grams (see Section 2.2). SMV^{light} has been applied, with a 50-50% train-development splitting: in this setting a linear kernel provided the best results.

Table 3. Experimental results for the Sentiment Analysis task

Unigrams	77.60%
POS tagged Unigrams	82.51%
Noisy POS 4-grams (no pre-proc.)	77.59%
POS 4-grams	**83.61%**
Unigrams [10]	82.20%
POS tagged Unigrams [10]	83.00%

As Table 3 suggests, the results improve on [10], as the adopted grammatical information is helpful. The test set employed in our experiments is slightly more complex, as the *Unigrams* model achieves a significantly worse result than in [10]. Moreover, without pre-processing, POS tags are inaccurate and this reflects in the lower performances of the Noisy POS 4-grams model. Our approach achieves a new state-of-art (i.e. 83.61%) on the dataset. This results due to the grammatical information provided by the POS n-grams and the contribution of the proposed pre-processing method is crucial. When no pre-processing is applied, the noise introduce by the POS-tagger would produce a consistent performance reduction, i.e. 77.59% vs 82.51%. Error analysis

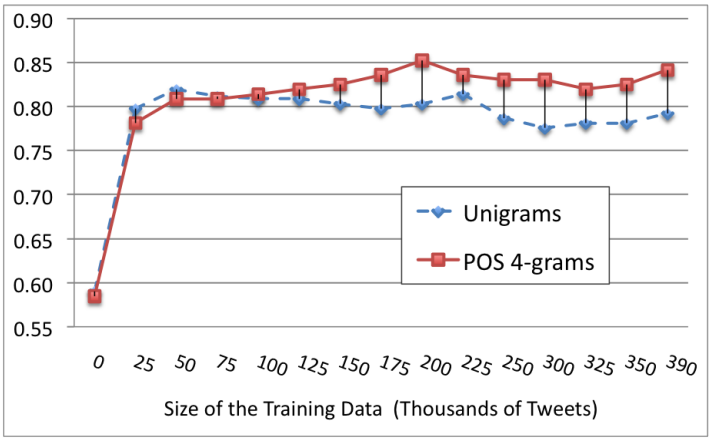


Fig. 2. Twitter Sentiment Analysis: accuracy

suggests that mistakes (e.g. the positive polarity given to the tweet "*Kobe is the best in the world not Lebron*") are due to lack of information. If LeBron James (and not Kobe) is the focus then the polarity is negative. But the alternative decision would have been perfectly acceptable, otherwise. Figure 2 reports the learning curve for the system with and without POS n-grams: POS n-grams are responsible of a faster convergence to higher accuracy levels.

4 Conclusions

In this paper shallow grammatical features as sequences of POS tags (i.e. POS n-grams) are proposed as a robust and effective model of grammatical information in different semantic tasks. Every experiment shows that state-of-the-art results are achieved or closely approximated by our modeling. Although standard training algorithms are here adopted, simple kernels over POS n-grams are quite effective, as for example the sentiment analysis tests demonstrate. Surprisingly, in Question Classification our model equals the accuracy of a performant tree kernel. The training complexity of the proposed models is very low. Although several optimization methods for tree kernel learning have been proposed (e.g. [6, 18]), our simpler approach is more applicable by posing much weaker requirements in terms of quality and size of the annotated datasets. This makes the proposed technology quite appealing for complex NLP and IR applications, such as the treatment of noisy sources that current micro-blogging trends require. This is also shown by the performances observed in the tweet sentiment analysis task, for which state-of-the-art results are obtained.

References

1. Baeza-Yates, R.A., Ribeiro-Neto, B.: Modern Information Retrieval. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA (1999)
2. Barbosa, L., Feng, J.: Robust sentiment detection on twitter from biased and noisy data. In: Coling 2010: Posters. pp. 36–44. Coling 2010 Organizing Committee, Beijing, China (August 2010)
3. Bilotti, M.W., Elsas, J.L., Carbonell, J., Nyberg, E.: Rank learning for factoid question answering with linguistic and semantic constraints. In: Proceedings of ACM CIKM (2010)
4. Collins, M.: Three generative, lexicalised models for statistical parsing. In: Proceedings of ACL 1997. pp. 16–23 (1997)
5. Collins, M., Duffy, N.: Convolution kernels for natural language. In: Proceedings of Neural Information Processing Systems (NIPS). pp. 625–632 (2001)
6. Croce, D., Moschitti, A., Basili, R.: Structured lexical similarity via convolution kernels on dependency trees. In: Proceedings of EMNLP. Edinburgh, Scotland, UK. (2011)
7. Foster, J., Özlem Çetinoğlu, Wagner, J., Roux, J.L., Hogan, S., Nivre, J., Hogan, D., van Genabith, J.: #hardtoparse: Pos tagging and parsing the twitterverse. In: Prooceedings of AAAI-11 Workshop on Analysing Microtext. San Francisco, CA (August 2011)
8. Freund, Y., Schapire, R.E.: Large margin classification using the perceptron algorithm. Machine Learning Journal 37(3), 277–296 (1999)
9. Gildea, D., Jurafsky, D.: Automatic Labeling of Semantic Roles. Computational Linguistics 28(3), 245–288 (2002)

10. Go, A., Bhayani, R., Huang, L.: Twitter Sentiment Classification using Distant Supervision. In: CS224N Project Report, Stanford (2009)
11. Joachims, T.: Text categorization with support vector machines: Learning with many relevant features. In: In Proceedings of the European Conference on Machine Learning (1998)
12. Johansson, R., Moschitti, A.: Extracting opinion expressions and their polarities – exploration of pipelines and joint models. In: Proceedings of ACL-HLT. Portland, Oregon, USA (2011)
13. Johansson, R., Nugues, P.: Dependency-based syntactic-semantic analysis with propbank and nombank. In: Proceedings of CoNLL-2008. Manchester, UK (August 16-17 2008)
14. Johansson, R., Nugues, P.: The effect of syntactic representation on semantic role labeling. In: Proceedings of COLING. Manchester, UK (August 18-22 2008)
15. Kaufmann, J., Kalita, J.: Syntactic normalization of twitter messages. In: International Conference on Natural Language Processing (2010)
16. Kwok, C.C.T., Etzioni, O., Weld, D.S.: Scaling question answering to the web. In: WWW. pp. 150–161 (2001)
17. Li, X., Roth, D.: Learning question classifiers. In: Proceedings of ACL'02 (2002)
18. Moschitti, A.: Efficient convolution kernels for dependency and constituent syntactic trees. In: ECML. pp. 318–329. Machine Learning: ECML 2006, 17th European Conference on Machine Learning, Proceedings, Berlin, Germany (September 2006)
19. Moschitti, A., Pighin, D., Basili, R.: Tree kernels for semantic role labeling. Computational Linguistics 34 (2008)
20. Moschitti, A., Quarteroni, S., Basili, R., Manandhar, S.: Exploiting syntactic and shallow semantic kernels for question answer classification. In: In Proc. of ACL-07. pp. 776–783 (2007)
21. Moschitti, A., Quarteroni, S., Basili, R., Manandhar, S.: Exploiting syntactic and shallow semantic kernels for question/answer classification. In: Proceedings of ACL'07 (2007)
22. Pang, B., Lee, L.: Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval 2(1-2), 1–135 (Jan 2008)
23. Tomás, D., Giuliano, C.: A semi-supervised approach to question classification. In: Proceedings of the 17th European Symposium on Artificial Neural Networks, Bruges, Belgium (2009)
24. Voorhees, E.M., Harman, D.: Overview of the seventh text retrieval conference trec-7. In: Proceedings of the Seventh Text REtrieval Conference (TREC-7. pp. 1–24 (1998)
25. Zanzotto, F.M., Pennacchiotti, M., Tsioutsiouliklis, K.: Linguistic redundancy in twitter. In: Proceedings of 2011 Conference on Empirical Methods on Natural Language Processing (EmNLP) (2011)

Encoding syntactic dependencies using Random Indexing and Wikipedia as a corpus

Pierpaolo Basile and Annalina Caputo

Dept. of Computer Science, University of Bari "Aldo Moro"
Via Orabona, 4, I-70125, Bari (ITALY)
{basilepp,acaputo}@di.uniba.it

Abstract. Distributional approaches are based on a simple hypothesis: the meaning of a word can be inferred from its usage. The application of that idea to the vector space model makes possible the construction of a WordSpace in which words are represented by mathematical points in a geometric space. Similar words are represented close in this space and the definition of "word usage" depends on the definition of the context used to build the space, which can be the whole document, the sentence in which the word occurs, a fixed window of words, or a specific syntactic context. However, in its original formulation WordSpace can take into account only one definition of context at a time. We propose an approach based on vector permutation and Random Indexing to encode several syntactic contexts in a single WordSpace. We adopt WaCkypedia_EN corpus to build our WordSpace that is a 2009 dump of the English Wikipedia (about 800 million tokens) annotated with syntactic information provided by a full dependency parser. The effectiveness of our approach is evaluated using the GEometrical Models of natural language Semantics (GEMS) 2011 Shared Evaluation data.

1 Background and motivation

Distributional approaches usually rely on the WordSpace model [20]. An overview can be found in [18]. This model is based on a vector space in which points are used to represent semantic concepts, such as words.

The core idea behind WordSpace is that words and concepts are represented by points in a mathematical space, and this representation is learned from text in such a way that concepts with similar or related meanings are near to one another in that space (geometric metaphor of meaning). The semantic similarity between concepts can be represented as proximity in an n-dimensional space. Therefore, the main feature of the geometric metaphor of meaning is not that meanings can be represented as locations in a semantic space, but rather that similarity between word meanings can be expressed in spatial terms, as proximity in a high-dimensional space.

One of the great virtues of WordSpaces is that they make very few language-specific assumptions, since just tokenized text is needed to build semantic spaces. Even more important is their independence from the quality (and the quantity)

of available training material, since they can be built by exploiting an entirely unsupervised distributional analysis of free text. Indeed, the basis of the WordSpace model is the *distributional hypothesis* [10], according to which the meaning of a word is determined by the set of textual *contexts* in which it appears. As a consequence, in distributional models words can be represented as vectors built over the observable *contexts*. This means that words are semantically related as much as they are represented by similar vectors. For example, if "basketball" and "tennis" occur frequently in the same context, say after "play", they are semantically related or similar according to the distributional hypothesis.

Since co-occurrence is defined with respect to a context, co-occurring words can be stored into matrices whose rows represent the terms and columns represent contexts. More specifically, each row corresponds to a vector representation of a word. The strength of the semantic association between words can be computed by using cosine similarity.

A weak point of distributional approaches is that they are able to encode only one definition of context at a time. The type of semantics represented in a WordSpace depends on the context. If we choose documents as context we obtain a semantics different from the one we would obtain by selecting sentences as context. Several approaches have investigated the aforementioned problem: [2] use a representation based on third-order tensors and provide a general framework for distributional semantics in which it is possible to represent several aspects of meaning using a single data structure. [19] adopt vector permutations as a means to encode order in WordSpace, as described in Section 2. BEAGLE [12] is a very well-known method to encode word order and context information in WordSpace. The drawback of BEAGLE is that it relies on a complex model to build vectors which is computational expensive. This problem is solved by [9] in which the authors propose an approach similar to BEAGLE, but using a method based on Circular Holographic Reduced Representations to compute vectors.

All these methods tackle the problem of representing word order in WordSpace, but they do not take into account syntactic context. A valuable attempt in this direction is described in [17]. In this work, the authors propose a method to build WordSpace using information about syntactic dependencies. In particular, they consider syntactic dependencies as context and assign different weights to each kind of dependency. Moreover, they take into account the distance between two words into the graph of dependencies. The results obtained by the authors support our hypothesis that syntactic information can be useful to produce effective WordSpace. Nonetheless, their methods are not able to directly encode syntactic dependencies into the space.

This work aims to provide a simple approach to encode syntactic relations dependencies directly into the WordSpace, dealing with both the scalability problem and the possibility to encode several context information. To achieve that goal, we developed a strategy based on Random Indexing and vector permutations. Moreover, this strategy opens new possibilities in the area of semantic composition as a result of the inherent capability of encoding relations between words.

The paper is structured as follows. Section 2 describes Random Indexing, the strategy for building our WordSpace, while details about the method used to encode syntactic dependencies are reported in Section 3. Section 4 describes a first attempt to define a model for semantic composition which relies on our WordSpace. Finally, the results of the evaluation performed using the GEMS 2011 Shared Evaluation data[1] is presented in Section 5, while conclusions are reported in Section 6.

2 Random Indexing

We exploit Random Indexing (RI), introduced by Kanerva [13], for creating a WordSpace. This technique allows us to build a WordSpace with no need for (either term-document or term-term) matrix factorization, because vectors are inferred by using an incremental strategy. Moreover, it allows to solve efficiently the problem of reducing dimensions, which is one of the key features used to uncover the "latent semantic dimensions" of a word distribution.

RI is based on the concept of Random Projection according to which high dimensional vectors chosen randomly are "nearly orthogonal".

Formally, given an $n \times m$ matrix A and an $m \times k$ matrix R made up of k m-dimensional random vectors, we define a new $n \times k$ matrix B as follows:

$$B^{n,k} = A^{n,m} \cdot R^{m,k} \quad k << m \tag{1}$$

The new matrix B has the property to preserve the distance between points. This property is known as Johnson-Lindenstrauss lemma: if the distance between two any points of A is d, then the distance d_r between the corresponding points in B will satisfy the property that $d_r = c \cdot d$. A proof of that property is reported in [8].

Specifically, RI creates a WordSpace in two steps (in this case we consider the document as context):

1. a context vector is assigned to each document. This vector is sparse, high-dimensional and ternary, which means that its elements can take values in {-1, 0, 1}. A context vector contains a small number of randomly distributed non-zero elements, and the structure of this vector follows the hypothesis behind the concept of Random Projection;
2. context vectors are accumulated by analyzing terms and documents in which terms occur. In particular, the semantic vector for a term is computed as the sum of the context vectors for the documents which contain that term. Context vectors are multiplied by term occurrences or other weighting functions, for example log-entropy.

Formally, given a collection of documents D whose vocabulary of terms is V (we denote with $dim(D)$ and $dim(V)$ the dimension of D and V, respectively) the above steps can be formalized as follows:

[1] Available on line:
http://sites.google.com/site/geometricalmodels/shared-evaluation

1. $\forall d_i \in D$, $i = 0, .., dim(D)$ we built the correspondent randomly generated context vector as:
$$\overrightarrow{r_j} = (r_{i1}, ..., r_{in}) \tag{2}$$
where $n \ll \dim(D)$, $r_{i*} \in \{-1, 0, 1\}$ and $\overrightarrow{r_j}$ contains only a small number of elements different from zero;
2. the WordSpace is made up of all term vectors $\overrightarrow{t_j}$ where:
$$\overrightarrow{t_j} = w_j \sum_{\substack{d_i \in D \\ t_j \in d_i}} \overrightarrow{r_i} \tag{3}$$
and w_j is the weight assigned to t_j in d_i.

By considering a fixed window W of terms as context, the WordSpace is built as follows:

1. a context vector is assigned to each term;
2. context vectors are accumulated by analyzing terms which co-occur in a window W. In particular, the semantic vector for each term is computed as the sum of the context vectors for terms which co-occur in W.

It is important to point out that the classical RI approach can handle only one context at a time, such as the whole document or the window W.

A method to add information about context (word order) in RI is proposed in [19]. The authors describe a strategy to encode word order in RI by permutation of coordinates in random vector. When the coordinates are shuffled using a random permutation, the resulting vector is nearly orthogonal to the original one. That operation corresponds to the generation of a new random vector. Moreover, by applying a predetermined mechanism to obtain random permutations, such as elements rotation, it is always possible to reconstruct the original vector using the reverse permutations. By exploiting this strategy it is possible to obtain different random vectors for each context[2] in which the term occurs. Let us consider the following example "The cat eats the mouse". To encode the word order for the word "cat" using a context window $W = 3$, we obtain:

$$\begin{aligned} < cat >= (\Pi^{-1} the) + (\Pi^{+1} eat) + \\ +(\Pi^{+2} the) + (\Pi^{+3} mouse) \end{aligned} \tag{4}$$

where $\Pi^n x$ indicates a rotation by n places of the elements in the vector x. Indeed, the rotation is performed by n right-shifting steps.

3 Encoding syntactic dependencies

Our idea is to encode syntactic dependencies, instead of words order, in the WordSpace using vector permutations.

[2] In the case in point the context corresponds to the word order

A syntactic dependency between two words is defined as:

$$dep(head, dependent) \tag{5}$$

where *dep* is the syntactic link which connects the *dependent* word to the *head* word. Generally speaking, *dependent* is the modifier, object or complement, while *head* plays a key role in determining the behavior of the link. For example, $subj(eat, cat)$ means that "cat" is the subject of "eat". In that case the *head* word is "eat", which plays the role of verb.

The key idea is to assign a permutation function to each kind of syntactic dependencies. Formally, let D be the set of all dependencies that we take into account. The function $f : D \rightarrow \Pi$ returns a schema of vector permutation for each $dep \in D$. Then, the method adopted to construct a semantic space that takes into account both syntactic dependencies and Random Indexing can be defined as follows:

1. a random context vector is assigned to each term, as described in Section 2 (Random Indexing);
2. random context vectors are accumulated by analyzing terms which are linked by a dependency. In particular the semantic vector for each term t_i is computed as the sum of the permuted context vectors for the terms t_j which are dependents of t_i and the inverse-permuted vectors for the terms t_j which are heads of t_i. The permutation is computed according to f. If $f(d) = \Pi^n$ the inverse-permutation is defined as $f^{-1}(d) = \Pi^{-n}$: the elements rotation is performed by n left-shifting steps.

Adding permuted vectors to the head word and inverse-permuted vectors to the corresponding dependent words allows to encode the information about both heads and dependents into the space. This approach is similar to the one investigated by [6] for encoding relations between medical terms.

To clarify, we provide an example. Given the following definition of f:

$$f(subj) = \Pi^{+3} \qquad f(obj) = \Pi^{+7} \tag{6}$$

and the sentence "The cat eats the mouse", we obtain the following dependencies:

$$\begin{array}{cc} det(the, cat) & subj(eat, cat) \\ obj(eat, mouse) & det(the, mouse) \end{array} \tag{7}$$

The semantic vector for each word is computed as:

– *eat*:

$$< eat >= (\Pi^{+3} cat) + (\Pi^{+7} mouse) \tag{8}$$

– *cat*:

$$< cat >= (\Pi^{-3} eat) \tag{9}$$

– *mouse*:

$$< mouse >= (\Pi^{-7} eat) \tag{10}$$

In the above examples, the function f does not consider the dependency *det*.

4 Compositional semantics

In this section we provide some initial ideas about semantic composition relying on our WordSpace. Distributional approaches represent words in isolation and they are typically used to compute similarities between words. They are not able to represent complex structures such as phrases or sentences. In some applications, such as Question Answering and Text Entailment, representing text by single words is not enough. These applications would benefit from the composition of words in more complex structures. The strength of our approach lies on the capability of codify syntactic relations between words overcoming the "word isolation" issue.

Recent work in compositional semantics argue that tensor product ($\otimes$) could be useful to combine word vectors. In [21] some preliminary investigations about product and tensor product are provided, while an interesting work by Clark and Pulman [5] proposes an approach to combine symbolic and distributional models. The main idea is to use tensor product to combine these two aspects, but the authors do not describe a method to represent symbolic features, such as syntactic dependencies. Conversely, our approach deals with symbolic features by encoding syntactic information directly into the distributional model. The authors in [5] propose a strategy to represent a sentence like "man reads magazine" by tensor product:

$$man \otimes subj \otimes read \otimes obj \otimes magazine \tag{11}$$

They also propose a solid model for compositionality, but they do not provide a strategy to represent symbolic relations, such as *subj* and *obj*. Indeed, they state: "How to obtain vectors for the dependency relations - subj, obj, etc. - is an open question". We believe that our approach can tackle this problem by encoding the dependency directly in the space, because each semantic vector in our space contains information about syntactic roles.

The representation based on tensor product is useful to compute sentence similarity. For example, given the previous sentence and the following one: "woman browses newspaper", we want to compute the similarity between those two sentences. The sentence "woman browses newspaper", using the compositional model, is represented by:

$$woman \otimes subj \otimes browse \otimes obj \otimes newspaper \tag{12}$$

Finally, we can compute the similarity between the two sentences by inner product, as follows:

$$(man \otimes subj \otimes read \otimes obj \otimes magazine) \cdot (woman \otimes subj \otimes browse \otimes obj \otimes newspaper) \tag{13}$$

Computing the similarity requires to calculate the tensor product between each sentence element and then compute the inner product. This task is complex, but exploiting the following property of the tensor product:

$$(w_1 \otimes w_2) \cdot (w_3 \otimes w_4) = (w_1 \cdot w_3) \times (w_2 \cdot w_4) \tag{14}$$

the similarity between two sentences can be computed by taking into account the pairs in each dependency and multiplying the inner products as follows:

$$man \cdot woman \times read \cdot browse \times \\ \times magazine \cdot newspaper \tag{15}$$

According to the property above mentioned, we can compute the similarity between sentences without using the tensor product. However, some open questions arise. This simple compositional strategy allows to compare sentences which have similar dependency trees. For example, the sentence "the dog bit the man" cannot can be compared to "the man was bitten by the dog". This problem can be easily solved by identifying active and passive forms of a verb. When two sentences have different trees, Clark and Pulman [5] propose to adopt the *convolution kernel* [11]. This strategy identifies all the possible ways of decomposing the two trees, and sums up the similarities between all the pairwise decompositions. It is important to point out that, in a more recent work, Clark et al. [4] propose a model based on [5] combined with a compositional theory for grammatical types, known as Lambek's pregroup semantics, which is able to take into account grammar structures. However, this strategy does not allow to encode grammatical roles into the WordSpace. This peculiarity makes our approach different. A more recent approach to distributional semantics and tree kernel can be found in [7] where authors propose a tree kernel that exploits distributional features to compute similarity between words.

5 Evaluation

The goal of the evaluation is to prove the capability of our approach in compositional semantics task exploiting the dataset proposed by Mitchell and Lapata [15], which is part of the "GEMS 2011 Shared Evaluation". The dataset is a list of two pairs of adjective-noun/verb-object combinations or compound nouns. Humans rated pairs of combinations according to similarity. The dataset contains 5,833 rates which range from 1 to 7. Examples of pairs follow:

```
support offer help provide 7
old person right hand 1
```

where the similarity between offer-support and provide-help (verb-object) is higher than the one between old-person and right-hand (adjective-noun). As suggested by the authors, the goal of the evaluation is to compare the system performance against humans scores by Spearman correlation.

5.1 System setup

The system is implemented in Java and relies on some portions of code publicly available in the Semantic Vectors package [22]. For the evaluation of the system, we build our WordSpaces using the WaCkypedia_EN corpus[3].

[3] Available on line: http://wacky.sslmit.unibo.it/doku.php?id=corpora

Dependency	Description	Permutation
OBJ	object of verbs	Π^{+7}
SBJ	subject of verbs	Π^{+3}
NMOD	the relationship between a noun and its adjunct modifier	Π^{+11}
COORD	coordination	Π^{+23}

Table 1. The set of dependencies used in the evaluation.

WaCkypedia_EN is based on a 2009 dump of the English Wikipedia (about 800 million tokens) and includes information about: PoS, lemma and a full dependency parse performed by MaltParser [16].

Our approach involves some parameters. We set the random vector dimension to 4,000 and the number of non-zero elements in the random vector equal to 10. We restrict the WordSpace to the 500,000 most frequent words. Another parameter is the set of dependencies that we take into account. In this preliminary investigation we consider the four dependencies described in Table 1 which reports also the kind of permutation[4] applied to each dependency.

5.2 Results

In this section, we provide the results of semantic composition. Table 2 reports the Spearman correlation between the output of our system and the scores given by the humans. Table 2 shows results for each type of combination: verb-object, adjective-noun and compound nouns. Moreover, Table 2 shows the results obtained when two other corpora were used for building the WordSpace: ukWaC [1] and TASA.

ukWaC contains 2 billion words and is constructed from the Web by limiting the crawling to the .uk domain and using medium-frequency words from the BNC corpus as seeds. We use only a portion of ukWaC corpus consisting of 7,025,587 sentences (about 220,000 documents).

The TASA corpus contains a collection of English texts that is approximately equivalent to what an average college-level student has read in his/her lifetime. More details about results on ukWaC and TASA corpora are reported in an our previous work [3].

It is important to underline that syntactic dependencies in ukWaC and TASA are extracted using MINIPAR[5] [14] instead of the MaltParser adopted by WaCkypedia_EN.

The results show that WaCkypedia_EN provides a significant improvement with respect to TASA and ukWaC. This result is mainly due to two factors: (1) the WordSpace built using WaCkypedia_EN contains more words and dependencies; (2) MaltParser produces more accurate dependencies than MINIPAR. However, considering adjective-noun relation, TASA corpus obtains the best re-

[4] The number of rotations is randomly chosen.

[5] MINIPAR is available at http://webdocs.cs.ualberta.ca/∼lindek/minipar.htm

Corpus	Combination	ρ
WaCkypedia_EN	**verb-object**	**0.257**
	adjective-noun	**0.346**
	compound nouns	**0.254**
	overall	**0.299**
TASA	verb-object	0.160
	adjective-noun	0.435
	compound nouns	0.243
	overall	0.186
ukWaC	verb-object	0.190
	adjective-noun	0.303
	compound nouns	0.159
	overall	0.179

Table 2. GEMS 2011 Shared Evaluation results.

sult and generally all corpora obtain their best performance in this relation. Probably, it is easier to discriminate this kind of relation than others.

Another important point, is that TASA corpus provides better results than ukWaC in spite of the huger number of relations encoded in ukWaC. We believe that texts in ukWaC contain more noise because they are extracted from the Web.

As future research, we plan to conduct an experiment similar to the one proposed in [15], which is based on the same dataset used in our evaluation. The idea is to use the composition functions proposed by the authors in our WordSpace, and compare them with our compositional model. In order to perform a fair evaluation, our WordSpace should be built from the BNC corpus. Nevertheless, the obtained results seem to be encouraging and the strength of our approach relies on the capability of capturing syntactic relations in a semantic space. We believe that the real advantage of our approach, that is the possibility to represent several syntactic relations, leaves some room for exploration.

6 Conclusions

In this work, we propose an approach to encode syntactic dependencies in WordSpace using vector permutations and Random Indexing. WordSpace is built relying on WaCkypedia_EN corpus extracted from English Wikipedia pages which contains information about syntactic dependencies. Moreover, we propose an early attempt to use that space for semantic composition of short phrases.

The evaluation using the GEMS 2011 shared dataset provides encouraging results, but we believe that there are open points which deserve more investigation. In future work, we have planned a deeper evaluation of our WordSpace and a more formal study about semantic composition.

References

1. Baroni, M., Bernardini, S., Ferraresi, A., Zanchetta, E.: The WaCky Wide Web: A collection of very large linguistically processed Web-crawled corpora. Language Resources and Evaluation 43(3), 209–226 (2009)
2. Baroni, M., Lenci, A.: Distributional memory: A general framework for corpus-based semantics. Computational Linguistics 36(4), 673–721 (2010)
3. Basile, P., Caputo, A., Semeraro, G.: Encoding syntactic dependencies by vector permutation. In: Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics. pp. 43–51. Association for Computational Linguistics, Edinburgh, UK (July 2011)
4. Clark, S., Coecke, B., Sadrzadeh, M.: A compositional distributional model of meaning. In: Proceedings of the Second Quantum Interaction Symposium (QI-2008). pp. 133–140 (2008)
5. Clark, S., Pulman, S.: Combining symbolic and distributional models of meaning. In: Proceedings of the AAAI Spring Symposium on Quantum Interaction. pp. 52–55 (2007)
6. Cohen, T., Widdows, D., Schvaneveldt, R., Rindflesch, T.: Logical leaps and quantum connectives: Forging paths through predication space. In: AAAI-Fall 2010 Symposium on Quantum Informatics for Cognitive, Social, and Semantic Processes. pp. 11–13 (2010)
7. Croce, D., Moschitti, A., Basili, R.: Structured lexical similarity via convolution kernels on dependency trees. In: Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing. pp. 1034–1046. Association for Computational Linguistics, Edinburgh, Scotland, UK. (July 2011)
8. Dasgupta, S., Gupta, A.: An elementary proof of the Johnson-Lindenstrauss lemma. Tech. rep., Technical Report TR-99-006, International Computer Science Institute, Berkeley, California, USA (1999)
9. De Vine, L., Bruza, P.: Semantic Oscillations: Encoding Context and Structure in Complex Valued Holographic Vectors. Quantum Informatics for Cognitive, Social, and Semantic Processes (QI 2010) (2010)
10. Harris, Z.: Mathematical Structures of Language. New York: Interscience (1968)
11. Haussler, D.: Convolution kernels on discrete structures. Technical Report UCSC-CRL-99-10 (1999)
12. Jones, M., Mewhort, D.: Representing word meaning and order information in a composite holographic lexicon. Psychological review 114(1), 1–37 (2007)
13. Kanerva, P.: Sparse Distributed Memory. MIT Press (1988)
14. Lin, D.: Dependency-based evaluation of MINIPAR. Treebanks: building and using parsed corpora (2003)
15. Mitchell, J., Lapata, M.: Composition in distributional models of semantics. Cognitive Science 34(8), 1388–1429 (2010)
16. Nivre, J., Hall, J., Nilsson, J.: Maltparser: A data-driven parser-generator for dependency parsing. In: Proceedings of LREC. vol. 6, pp. 2216–2219 (2006)
17. Padó, S., Lapata, M.: Dependency-based construction of semantic space models. Computational Linguistics 33(2), 161–199 (2007)
18. Sahlgren, M.: The Word-Space Model: Using distributional analysis to represent syntagmatic and paradigmatic relations between words in high-dimensional vector spaces. Ph.D. thesis, Stockholm: Stockholm University, Faculty of Humanities, Department of Linguistics (2006)

19. Sahlgren, M., Holst, A., Kanerva, P.: Permutations as a means to encode order in word space. In: Proceedings of the 30th Annual Meeting of the Cognitive Science Society (CogSci'08) (2008)
20. Schütze, H.: Word space. In: Hanson, S.J., Cowan, J.D., Giles, C.L. (eds.) Advances in Neural Information Processing Systems. pp. 895–902. Morgan Kaufmann Publishers (1993)
21. Widdows, D.: Semantic vector products: Some initial investigations. In: The Second AAAI Symposium on Quantum Interaction (2008)
22. Widdows, D., Ferraro, K.: Semantic Vectors: A Scalable Open Source Package and Online Technology Management Application. In: Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC 2008) (2008)

Algebraic compositional models for semantic similarity in ranking and clustering

Paolo Annesi, Valerio Storch, Danilo Croce and Roberto Basili

Dept. of Computer Science,
University of Roma Tor Vergata, Roma, Italy
{annesi,croce,basili}@info.uniroma2.it
storch@uniroma2.it

Abstract. Although distributional models of word meaning have been widely used in Information Retrieval achieving an effective representation and generalization schema of words in isolation, the composition of words in phrases or sentences is still a challenging task. Different methods have been proposed to account on syntactic structures to combine words in term of algebraic operators (e.g. tensor product) among vectors that represent lexical constituents.
In this paper, a novel approach for semantic composition based on space projection techniques over the basic geometric lexical representations is proposed. In the geometric perspective here pursued, syntactic bi-grams are projected in the so called *Support Subspace*, aimed at emphasizing the semantic features shared by the compound words and better capturing phrase-specific aspects of the involved lexical meanings. State-of-the-art results are achieved in a well known benchmark for phrase similarity task and the generalization capability of the proposed operators is investigated in a cross-linguistic scenario, i.e. in the English and Italian Language.

1 Introduction

With the rapid development of the World Wide Web and the spread of human-generated contents, Information Retrieval (IR) has many challenges in discovering and exploiting those rich and huge information resources. Semantic search [3] improves search precision and recall by understanding user's intent and the contextual meaning of concepts in documents and queries. Semantic search extends the scope of traditional information retrieval paradigms from mere document retrieval to entity and knowledge retrieval, improving the conventional IR methods by looking at a different perspective, i.e. the meaning of words. However, the language richness and its intrinsic relation to the world and human activities make semantic search a very complex task. In a IR system, a user can express its specific user need with a natural language query like "*... buy a car ...*". This request can be satisfied by documents expressing the abstract concept of *buying something* and in particular the focus of the action is a car. This information

can be expressed inside a document collection in many different forms, e.g. the quasi-synonymic expression "*... purchase an automobile ...*". Accounting on lexical overlap with respect to the original query, a Bag-of-word based system would instead retrieve different documents, containing expressions such as "*... buy a bag ...*" or "*... drive a car ...*". A proper semantic generalization is thus needed, in order to derive the correct *composition* of the target words, i.e. an action like *buy* and an object like *car*.

While compositional approaches to language understanding have been largely adopted, semantic tasks are still challenging for research in Natural Language Processing. Traditional logic-based approaches (as the Montague's approach in [17] and [2]) rely on Frege's principle for which the meaning of a sentence is a function of the meanings of its parts [10]. The resulting theory allows an algebra on the discrete propositional symbols to represent the meaning of arbitrarily complex expressions. Despite the fact that they are formally well defined, logic-based approaches have limitations in the treatment of ambiguity, vagueness and cognitive aspects intrinsically connected to natural language.

On the other hand, distributional models early introduced by Schütze [21] rely on the Word Space model. Here semantic uncertainty is managed through the statistical analysis of large scale corpora. Linguistic phenomena are then modeled according to a geometrical perspective, i.e. points in a high-dimensional space representing semantic concepts, such as words, and can be learned from corpora, in such a way that similar, or related, concepts are near each another in the space. Methods for constructing representations for phrases or sentences through vector composition has recently received a wide attention in literature (e.g. [15, 23]). However, vector-based models typically represent isolated words and ignore grammatical structure [23]. Such models have thus a limited capability to model compositional operations over phrases and sentences.

In order to overcome these limitations a so-called compositional distributional semantics (DCS) model is needed and its development is still object of on-going and controversial research (e.g. [5], [11]). A compositional model based on distributional analysis should provide semantic information consistent with the meaning assignment that is typical of human subjects. For example, it should support synonymy and similarity judgments on phrases, rather than only on single words. The objective should be a measure of similarity between quasi-synonymic complex expressions, such as "*... buy a car ...*" vs. "*... purchase an automobile ...*". Another typical benefit should be a computational model for entailment, so that the representation for " *... buying something ...*" should be implied by the expression "*... buying a car ...*" but not by "*... buying time ...*". Distributional compositional semantics (DCS) need thus a method to define: (1) a way to represent lexical vectors $\boldsymbol{u}$ and $\boldsymbol{v}$, for words u, v dependent on the phrase (r, u, v) (where r is a syntactic relation, such as verb-object), and (2) a metric for comparing different phrases according to the selected representations $\boldsymbol{u}$, $\boldsymbol{v}$. Existing models are still controversial and provide general algebraic operators (such as tensor products) over lexical vectors.

In this paper, we focus on the geometry of latent semantic spaces by proposing a novel distributional model for semantic composition. The aim is to model semantics of syntactic bigrams as projections in lexically-driven subspaces. Distances in such subspaces (called *Support Spaces*) emphasize the role of *common* features that constraint in "parallel" the interpretation of the involved lexical meanings and better capture phrase-specific aspects. In the following evaluations, operators will be employed to compose word pairs involved in specific syntactic structures. This resulting compositions will be evaluated according two different perspectives. First, similarity among compositions will be evaluated with respect to human annotators' judgments. Then, the operators generalization capability will be measured in order to prove their applicability in semantic search complex systems. Moreover the robustness of this Support Spaces based will be confirmed in a cross-linguistic scenario, i.e. in the English and Italian Language.

While Section 2 discusses existing methods of compositional distributional semantics, Section 3 presents our model based on support spaces. Experiments in Section 4 are used to show the beneficial impact of the proposed model and the contribution to semantic search systems. Finally, Section 5 derives the conclusions.

2 Related work

While compositional semantics allows to govern the recursive interpretation of sentences or phrases, traditional vector space models (as in IR [20]) and, mostly, semantic space models, such as LSA ([7, 13]), represent lexical information in metric spaces where individual words are represented according to the distributional analysis of their co-occurrences over a large corpus. Such models are based on the distributional hypotesis which assumes that words occurring within similar contexts are semantically similar (Harris in [12]).

Semantic spaces have been widely used for representing the meaning of words or other lexical entities (e.g. [23]), with successful applications in lexical disambiguation ([22]) or harvesting thesauri (as in Lin [14]). In this work we will refer to the so-called **word-based spaces**, in which words are represented by probabilistic information of their co-occurences calculated in a fixed range window over all sentences. In such models, vector components correspond to the entries f of the vocabulary V (i.e. to features that are individual words). Weigths are associated with each component, using different estimators of their correlation. In some works (e.g. [15]) pure co-occurrence counts are adopted as weighting functions f_i, where $i = 1, ..., N$ and $N = |V|$; in other works (e.g. [18]), statistical functions like the pointwise mutual information between the target word w and the captured co-occurences in the window are used, i.e. $pmi(w, i) = log_2 \frac{p(w,f_i)}{p(w)\cdot p(f_i)}$.

A vector $\boldsymbol{w} = (pmi_1, ..., pmi_N)$ models a word w and it is thus built over all the words f_i belonging to the dictionary. When w and f never co-occur in any window their pmi is by default set to 0. Weights of vector components depend on the size of the co-occurrence window and express the global statistics in the entire corpus. Larger values of the adopted window size aim to capture *topical*

similarity (as in the document based models of IR), while smaller sizes (usually between the ±1-3 surrounding words) lead to representation better suited for *paradigmatic similarities* between word vectors $\boldsymbol{w}$. Cosine similarity between vectors $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$ is modeled as the normalized scalar product, i.e. $\frac{\langle \boldsymbol{w}_1, \boldsymbol{w}_2 \rangle}{\|\boldsymbol{w}_1\| \|\boldsymbol{w}_2\|}$ that expresses *topical* or *paradigmatic similarity* according to the different representations (e.g. window sizes). Notice that dimensionality reduction methods, such as LSA [7, 13] are also applied in some studies, to capture second order dependencies between features f, i.e. applying semantic smoothing to possibly sparse input data. Applications of an LSA-based representation to Frame Induction or Semantic Role Labeling are presented in [19] and [6], respectively.

The main limitation of distributional models of lexical semantic is their non-compositional nature: they are based on statistics related to the occurences of the individual words in the corpus. In such models, the semantic of topological similarity functions is thus defined only for the comparison between individual words. That is the reason why distributional methods can not compute the meanings of phrases (and sentences) as effectively as they do indeed over individual words. Distributional methods have been recently extended to better account compositionality, in the so called distributional compositional semantics (DCS) approaches. Mitchell and Lapata in [15] follow Foltz [9] and assume that the contribution of the syntactic structure can be ignored, while the meaning of a phrase is simply the *commutative sum of the meanings of its constituent words*. More formally, [15] defines the composition $\boldsymbol{p}^{\circ} = \boldsymbol{u} \circ \boldsymbol{v}$ of vectors $\boldsymbol{u}$ and $\boldsymbol{v}$ through an additive class of composition functions expressed by:

$$\boldsymbol{p}^{+} = \boldsymbol{u} + \boldsymbol{v} \tag{1}$$

This perspective clearly leads to a variety of efficient yet shallow models of compositional semantics compared in [15]. For example pointwise multiplication is defined by the multiplicative function:

$$\boldsymbol{p}^{\cdot} = \boldsymbol{u} \odot \boldsymbol{v} \tag{2}$$

where the symbol $\odot$ represents multiplication of the corresponding components, i.e. $p_i = u_i \cdot v_i$. Point-wise multiplication seems to best correspond with the intended effects of syntactic interaction, as experiments in [15] demonstrate. In [8], the concept of a *structured vector space* is introduced, where each word is associated with a set of vectors corresponding to different syntactic dependencies. Every word is thus expressed by a tensor, and tensor operations are imposed.

The main differences among these studies lies in (1) the lexical vector representation selected (e.g. some authors do not even commit to any representation, but generically refer to any lexical vector, as in [11]) as well as in (2) the adopted compositional algebra, i.e. the system of operators defined over such vectors. Generally, proposed operators do not depend on the involved lexical items, but a general purpose algebra is adopted. Since compositional structures are highly lexicalized, and the same syntactic relation triggers to very different semantic relations with respect to the different involved words, a proposal that makes the compositionality operators dependent on individual lexical vectors is hereafter discussed.

3 A quantitative model for compositionality

In order to determine the semantic analogies and differences between two phrases, such as "*... buy a car ...*" and "*... buy time ...*", a distributional compositional model is employed as follows. The involved lexicals are *buy*, *car* and *time*, while their corresponding vector representation will be denoted by $\boldsymbol{w}_{buy}$ $\boldsymbol{w}_{car}$ and $\boldsymbol{w}_{time}$. The major result of most studies on DCS is the definition of the function $\circ$ that associates with $\boldsymbol{w}_{buy}$ and $\boldsymbol{w}_{car}$ a new vector $\boldsymbol{w}_{buy\_car} = \boldsymbol{w}_{buy} \circ \boldsymbol{w}_{car}$.

We consider this approach misleading since vector components in the word space are tied to the syntactic nature of the composed words and the new vector $\boldsymbol{w}_{buy\_car}$ should not have the same type of the original vectors. Notice also that the components of $\boldsymbol{w}_{buy}$ and $\boldsymbol{w}_{car}$ express all their contexts, i.e. interpretations, and thus senses, of *buy* and *car* in the corpus. Algebric operations are thus open to misleading contributions, brought by not-null feature scores of buy_i vs. car_j $(i \neq j)$ that may correspond to senses of *buy* and *car* that are not related to the specific phrase "*buy a car*". On the contrary, in a composition, such as the verb-object pair (buy, car), the word *car* influences the interpretation of the verb *buy* and viceversa. The model here proposed is based on the assumption that this influence can be expressed via the operation of projection into a subspace, i.e. a subset of original features f_i. A projection is a mapping (a selection function) over the set of all features. A subspace generated by a projection function Π local to the (buy, car) phrase can be found such that only the features specific to the phrase meaning are selected and the irrelevant ones are neglected. The resulting subspace has to preserve the compositional semantics of the phrase and it is called **support subspace** of the underlying word pair.

Consider the bigram composed of the words *buy* and *car* and their vectorial representation in a co-occurrence $N-$dimensional Word Space. Table 1 reports the $k = 10$ features with the highest contributions of the point wise product of the pairs *(buy,car)* and *(buy,time)*. The support space thus selects the most important features for both words, e.g. *buy*.V and *car*.N. Notice that this captures the conjunctive nature of the scalar product to which contributions come from feature with non zero scores in both vectors. It is clear that the two pairs give rise to different support subspaces: the main components related with *buy car* refer mostly to the automobile commerce area unlike the ones related with *buy time* mostly referring to the time wasting or saving. Similarity judgments about a pair can be thus better computed within its support subspace.

Buy-Car	Buy-Time
cheap::Adj	*consume::V*
insurance::N	*enough::Adj*
rent::V	*waste::V*
lease::V	*save::In*
dealer::N	*permit::N*
motorcycle::N	*stressful::Adj*
hire::V	*spare::Adj*
auto::N	*save::V*
california::Adj	*warner::N*
tesco::N	*expensive::Adj*

Table 1. Features corresponding to dimensions in the k=10 dimensional support space of bigrams *buy car* and *buy time*

More formally $k-$dimensional support subspace for a word pair (u, v) (with $k \ll N$) is the subspace spanned by the subset of $n \leq k$ indexes $\mathbf{I}^k(\boldsymbol{u}, \boldsymbol{v}) = \{i_1, ..., i_n\}$ for which $\sum_{t=1}^{n} u_{i_t} \cdot v_{i_t}$ is maximal. Given two pairs the similarity

between syntactic equivalent words (e.g. nouns with nouns, verbs with verbs) is measured in the support subspace derived by applying a specific projection function. Compositional similarity between *buy car* and the latter pairs (e.g. *buy time*) is thus estimated by (1) immersing w_{buy} and w_{time} in the selected "*... buy car ...*" support subspace and (2) estimating similarity between corresponding arguments of the pairs locally in that subspace. Therefore the similarity between syntactic equivalent words (e.g. *car* with *time*) within these new subspace is measured.

Therefore given a pair (u, v), a unique matrix $\mathbf{M}^k_{uv} = (m^k_{uv})_{ij}$ is defined for a given projection $\Pi^k(u, v)$ into the k-dimensional support space of any pair (u, v) according to the following definition:

$$(m^k_{uv})_{ij} = \begin{cases} 1 & \text{iff } i = j \in \mathbf{I}^k(\boldsymbol{u}, \boldsymbol{v}) \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

The vector $\tilde{\boldsymbol{u}}$ projected in the support subspace can be thus estimated through the following matrix operation:

$$\tilde{\boldsymbol{u}} = \Pi^k(u, v) \qquad \tilde{\boldsymbol{u}} = \mathbf{M}^k_{uv}\boldsymbol{u} \tag{4}$$

A special case of the projection matrix is given when no k limitation is imposed to the dimension and all the positive addends in the scalar product are taken. Notice also that two pairs $p_1 = (u, v)$ and $p_2 = (u', v')$ give rise to two different projections denoted by $\mathbf{M}^k_1$ and $\mathbf{M}^k_2$ and defined as:

$$(\textit{Left} \text{ projection}) \ \Pi^k_1 = \Pi^k(\boldsymbol{u}, \boldsymbol{v}) \qquad (\textit{Right} \text{ projection}) \ \Pi^k_2 = \Pi^k(\boldsymbol{u}', \boldsymbol{v}') \tag{5}$$

It is also possible to define a unique symmetric projection Π^k_{12} corresponding to the combined matrix $\mathbf{M}^k_{12}$ as follows:

$$\mathbf{M}^k_{12} = (\mathbf{M}^k_1 + \mathbf{M}^k_2) - (\mathbf{M}^k_1\mathbf{M}^k_2) \tag{6}$$

where the mutual components that satisfy Eq. 3 are employed as $\mathbf{M}^k_{12}$.
As Π_1 is the projection in the support subspace for the pair p_1, it is possible to immerse the latter pair p_2 by applying Eq. 4. **This results in the two vectors $\mathbf{M}^k_1\boldsymbol{u}'$ and the $\mathbf{M}^k_1\boldsymbol{v}'$.** It follows that a compositional similarity judgment between two phrase over the first pair support subspace can be expressed as:

$$\Phi^{(\circ)}_{p_1}(p_1, p_2) = \Phi^{(\circ)}_1(p_1, p_2) = \frac{\langle \mathbf{M}^k_1\boldsymbol{u}, \mathbf{M}^k_1\boldsymbol{u}' \rangle}{\left\|\mathbf{M}^k_1\boldsymbol{u}\right\| \left\|\mathbf{M}^k_1\mathbf{u}'\right\|} \circ \frac{\langle \mathbf{M}^k_1\boldsymbol{v}, \mathbf{M}^k_1\boldsymbol{v}' \rangle}{\left\|\mathbf{M}^k_1\boldsymbol{v}\right\| \left\|\mathbf{M}^k_1\boldsymbol{v}'\right\|} \tag{7}$$

where first cosine similarity between syntactically correlated vectors in the selected support subspaces are computed and then a composition function $\circ$, such as the sum or the product, is applied. Compositional function over the latter support subspace evoked by the pair p_2 can be correspondingly denoted by $\Phi^{(\circ)}_2(p_1, p_2)$. A symmetric composition function can thus be obtained as a combination of $\Phi^{(\circ)}_1(p_1, p_2)$ and $\Phi^{(\circ)}_2(p_1, p_2)$ as:

$$\Phi_{12}^{(\circ)}(p_1, p_2) = \Phi_1^{(\circ)}(p_1, p_2) \diamond \Phi_2^{(\circ)}(p_1, p_2) \quad (8)$$

where the composition function $\diamond$ (again the sum or the product) between the similarities over the left and right support subspaces is applied. Notice how the left and right composition operators ($\circ$) may differ from the overall composition operator $\diamond$. More details are discussed in [1].

4 Experimental Evaluation

This experimental evaluation aims to estimate the effectiveness of the proposed class of projection based methods in capturing similarity judgments over phrases and syntactic structures. In particular, a first evaluation is carried out to measure the correlation of the operator outcomes with judgments provided by human annotators. The generalization capability of the operators is measured in the second evaluation in order to prove their applicability in semantic search complex systems. Moreover the latter experiments are carried out in a cross-language setting, i.e. for english and italian datasets.

Type	First Pair	Second Pair	Rate
VO	*support offer*	*provide help*	7
	use knowledge	*exercise influence*	5
	achieve end	*close eye*	1
AdjN	*old person*	*right hand*	1
	vast amount	*large quantity*	7
	economic problem	*practical difficulty*	3
NN	*tax charge*	*interest rate*	7
	tax credit	*wage increase*	5
	bedroom window	*education officer*	1

Table 2. Example of Mitchell and Lapata dataset for the three syntactic relations verb-object (VO), adjective-noun (AdjN) and noun-noun (NN)

Two different word space are derived for the different languages. For English, the word space is derived from the ukWak [4], a web-based corpus consisting of about 2 billion tokens. For Italian, the Italian Wikipedia corpus[1] has been employed. It consists of about 200 million tokens and more than 10 million sentences. The space construction proceeds from an adjacency matrix M on which Singular Values decomposition ([7]) is then applied. Part-of-speech tagged words have been collected from the corpus to reduce data sparseness. Then all target words tws occurring more than 200 times are selected, i.e. more that 50,000 candidate features. Each column i of M represents a word w in the corpus. Rows model the target words tw, i.e. contain the pmi values for the individual features f_i, as captured in a window of size ± 3 around tw. The most frequent 20,000 left and right features f_i are selected, so that M expresses 40,000 contexts. SVD is here applied to limit dimensionality to $N = 100$.

4.1 Experiment I

The first evaluation is carried out over the dataset proposed by [16], which is part of the *GEMS 2011 Shared Evaluation*. It consists of a list of 5,833 adjective-noun (AdjN), verb-object (VO) or noun-noun (NN) pairs, rated with scores ranging from

[1] The corpus is developed by the WaCky community and it is available in the Wacky project web page at http://medialab.di.unipi.it/Project/QA/wikiCoNLL.bz2

1 to 7. In Table 2, examples of pairs and scores are shown. The correlation of the similarity judgements outputed by a DCS model against the human judgements is computed using Spearman's ρ, a non-parametric measure of statistical dependence between two variables proposed by [15].

Model		AdjN	NN	VO
Mitchell&Lapata Word Space SVD	Additive	.69	.70	**.64**
	Multiplicative	.38	.43	.42
Support Subspace[1]	$\Phi^{(+)}$, Π^k_{12} (k=30)	**.70**	**.71**	.63
	$\Phi^{(\cdot)}_{12}$, $\Phi^{(+)}_i$, Π^k_i (k=40)	.68	.68	**.64**
Agreement among Human Subjects	Max	.88	.92	.88
	Avg	.72	.72	.71

Table 3. Spearman's ρ correlation coefficients across Mitchell and Lapata models and the projection-based models proposed in Section 3. Word space refers to the source spaces used as input to the LSA decomposition model.

Table 3 reports M&L performances in the first row, while in the last row the max and the average interannotator agreement scores for the three categories derived through a leave one-out resampling method are shown. Row 2 shows Speraman's correlation for support subspace models discussed in [1] that better perform the distributional compositional task. Notice that different configurations according to the models described in Section 3 are used. For example, the system denoted as $\Phi^{(\cdot)}_{12}$, $\Phi^{(+)}_i$, Π^k_i (k=40), corresponds to a multiplicative symmetric composition function $\Phi^{(\cdot)}_{12}$ (as for Eq. 8) based on left and right additive compositions $\Phi^{(+)}_i$ ($i = 1, 2$ as in Eq. 7), derived through a projection Π^k_i in the support space limited to the first $k = 40$ components for each pair (as for Eq. 5). The specific operator denoted by $\Phi^{(+)}$, Π^k_{12} (k=30) achieves the best performance over two out of three syntactic patterns (i.e. AdjN and NN) and is close to the best figures for VO. Experimental evaluation shows that the best performances are achieved by the projection based operators proposed. Notice that the distributional composition between verbs and objects is a very tricky task and results are in line with the additive model. Globally the results of our models are close to the average agreement among human subjects, this latter representing a sort of upper bound for the underlying task. It seems that latent topics (as extracted through SVD from sentence and word spaces) as well as the projections operators defined by support subspaces, provide a suitable comprehensive paradigm for compositionality. They seem to capture compositional similarity judgements that are significantly close to human ones.
Notice that different settings of the projection operations can influence the performances. A more exhaustive study of the possible settings is presented in [1].

4.2 Experiment II

In this second evaluation, the generalization capability of the employed operators will be investigated. A verb (e.g. *perform*) can be more or less semantically

close to another verb (e.g. other verbs like *solve*, or *produce*) depending on the context in which it appears. The verb-object (`VO`) composition specifies the verb's meaning by expressing one of its selectional preferences, i.e. its object. In this scenario, we expect that a pair such as *perform task* will be more similar to *solve issue*, as they both reflect an abstract cognitive action, with respect to a pair like *produce car*, i.e. a concrete production. This kind of generalization capability is crucial to effectively use this class of operators in a QA scenario by enabling to rank results according to the complex representations of the question. Moreover, both English and Italian languages can be considered to demonstrate the impact in a cross language setting. Figure 4 shows a manually developed dataset. It consists of 24 `VO` word pairs in English and Italian, divided into 3 different semantic classes: `Cognitive`, `Ingest Liquid` and `Fabricate`.

Semantic Class	English	Italian
`Cognitive`	*perform task*	*svolgere compito*
	solve issue	*risolvere questione*
	handle problem	*gestire problema*
	use method	*applicare metodo*
	suggest idea	*suggerire idea*
	determine solution	*trovare soluzione*
	spread knowledge	*divulgare conoscenza*
	start argument	*iniziare ragionamento*
`Ingest Liquid`	*drink water*	*bere acqua*
	ingest syrup	*ingerire sciroppo*
	pour beer	*versare birra*
	swallow saliva	*inghiottire saliva*
	assume alcohol	*assumere alcool*
	taste wine	*assaggiare vino*
	sip liquor	*assaporare liquore*
	take coffee	*prendere caff*
`Fabricate`	*produce car*	*produrre auto*
	complete construction	*completare costruzione*
	fabricate toy	*fabbricare giocattolo*
	build tower	*edificare torre*
	assemble device	*assemblare dispositivo*
	construct building	*costruire edificio*
	manufacture product	*realizzare prodotto*
	create artwork	*creare opera*

Table 4. Cross-linguistic dataset

This evaluation aims to measure how the proposed compositional operators group together semantically related word pairs, i.e. those belonging to the same class, and separate the unrelated pairs. Figure 1 shows the application of two models, the Additive (eq. 1) and Support Subspace (Eq. 8) ones that achieve the best results in the previous experiment. The two languages are reported in different rows. Similarity distribution between the geometric representation of verb pair, with no composition, has been investigated as a baseline. For each language, the similarity distribution among the possible 552 verb pairs is estimated and two distributions of the `infra` and `intra-class` pairs are independently plotted. In order to summarize them, a Normal Distribution $N(\mu, \sigma^2)$ of mean μ and variance σ^2 are employed. Each point represents the percentage $p(x)$ of pairs in a group that have a given similarity value equal to x. In a given class, the `VO-VO` pairs of a DCS operator are expected to increase this probability with respect to the baseline pairs `V-V` of the same set. Viceversa, for pairs belonging to different classes, i.e. `intra-class` pairs. The distributions for the baseline

control set (i.e. `Verbs Only, V-V`) are always depicted by dotted lines, while DCS operators are expressed in continuous line.

Notice that the overlap between the curves of the `infra` and `intra-class` pairs corresponds to the amount of **ambiguity** in deciding if a pair is in the same class. It is the *error probability*, i.e. the percentage of cases of one group that by chance appears to have more probability in the other group. Although the actions described by different classes are very different, e.g. `Ingest Liquid` vs. `Fabricate`, most verbs are ambiguous: contextual information is expected to enable the correct decision. For example, although the class `Ingest Liquid` is clearly separated with respect to the others, a verb like *assume* could well be classified in the `Cognitive` class, as in *assume a position*.

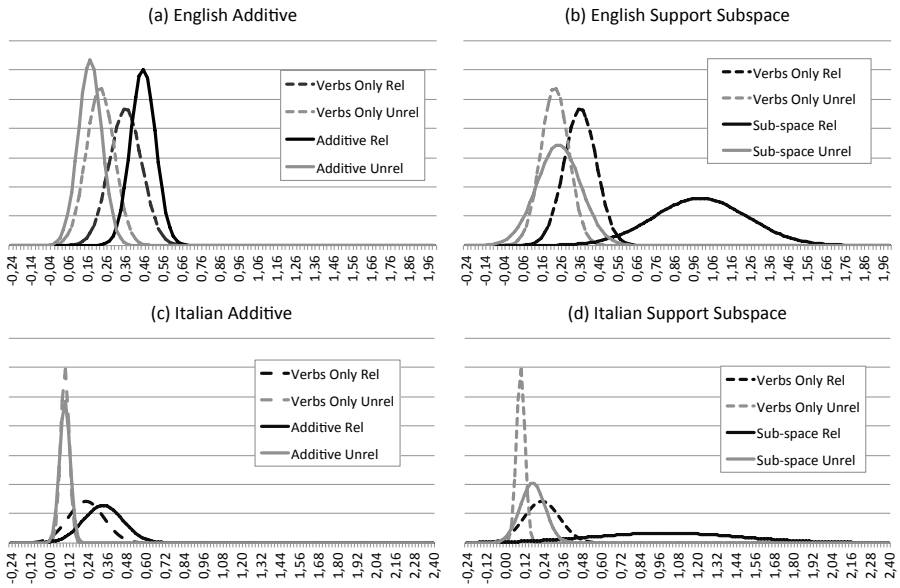


Fig. 1. Cross-linguistic Gaussian distribution of infra (red) and inter (green) clusters of the proposed operators (continous line) with respect to verbs only operator (dashed line)

The outcome of the experiment is that DCS operators are always able to increase the gap in the average similarity of the `infra` vs. `intra-class` pairs. It seems that the geometrical representation of the verb is consistently changed as most similarity distributions suggest. The compositional operators seem able to decrease the overlap between different distributions, i.e. reduce the ambiguity.

Figure 1 (a) and (c) report the distribution of the ML additive operator, that achieves an impressive ambiguity reduction, i.e. the overlap between curves is drastically reduced. This phenomenon is further increased when the Support

Subspace operator is employed as shown in Figure 1 (b) and (d): notice how the mean value of the distribution of semantically related word is significantly increased for both languages.

The probability of error reduction can be computed against the control groups. It is the decrease of the error probability of a DCS relative to the same estimate for the control (i.e. `V-V`) group. It is a natural estimator of the generalization capability of the involved operators. In Table 5 the intersection area for all the models and the decrement of the relative probability of error are shown. For English, the ambiguity reduction of the Support Subspace operator is of 91% with respect to the control set. This is comparable with the additive operator results, i.e. 92.3%. It confirms the findings of the previous experiment where the difference between these operators is negligible. For Italian, the generalization capability of support subspace operator is more stable, as its error reduction is of 62.9% with respect to the additive model, i.e. 54.2%.

	English		Italian	
Model	Probability of Error	Ambiguity Decrease	Probability of Error	Ambiguity Decrease
`VerbOnly`	.401	-	.222	-
`Additive`	.030	92.3%	.101	54.2%
`SupportSubspace`	.036	91.0%	.082	62.9%

Table 5. Ambiguity reduction analysis

5 Conclusions

In this paper, a distributional compositional semantic model based on space projection guided by syntagmatically related lexical pairs is defined. Syntactic bi-grams are here projected in the so called *Support Subspace* and compositional similarity scores are correspondingly derived. This represents a novel perspective on compositional models over vector representations with respect to shallow vector operators (e.g. additive or multiplicative operations) as proposed in literature, e.g. in [16]. The presented approach focuses on selecting the most important components for a specific word pair involved in a syntactic relation in order to have a more accurate estimator of their similarity.

The proposed method have been evaluated over the well known dataset in [16] achieving results close to the average human interannotator agreement scores. A first applicability study of such compositional models in typical IR systems was carried out. The operators' generalization capability was measured proving that compositional operators can effectively separate phrase structure in different semantic clusters. The robustness of such operators has been also confirmed in a cross-linguistic scenario, i.e. in the English and Italian Language. Future work on other compositional prediction tasks (e.g. selectional preference modeling) and over different datasets will be carried out to better assess and generalize the presented results.

References

1. Annesi, P., Storch, V., Basili, R.: Space projections as distributional models for semantic composition (2012), submitted for publication

2. B. Coecke, M.S., Clark, S.: Mathematical foundations for a compositional distributed model of meaning. Lambek Festschirft, Linguistic Analysis, vol. 36 36 (2010), http://arxiv.org/submit/10256/preview
3. Baeza-Yates, R., Ciaramita, M., Mika, P., Zaragoza, H.: Towards semantic search. Natural Language and Information Systems pp. 4–11 (2008)
4. Baroni, M., Bernardini, S., Ferraresi, A., Zanchetta, E.: The wacky wide web: a collection of very large linguistically processed web-crawled corpora. Language Resources And Evaluation 43(3), 209–226 (2009)
5. Baroni, M., Zamparelli, R.: Nouns are vectors, adjectives are matrices: representing adjective-noun constructions in semantic space. In: Proceedings of EMNLP 2010. pp. 1183–1193. EMNLP '10, Stroudsburg, PA, USA (2010)
6. Croce, D., Giannone, C., Annesi, P., Basili, R.: Towards open-domain semantic role labeling. In: ACL. pp. 237–246 (2010)
7. Deerwester, S.C., Dumais, S.T., Landauer, T.K., Furnas, G.W., Harshman, R.A.: Indexing by latent semantic analysis. JASIS 41(6), 391–407 (1990)
8. Erk, K., Pad, S.: A structured vector space model for word meaning in context (2008)
9. Foltz, P.W., Kintsch, W., Landauer, T.K., L, T.K.: The measurement of textual coherence with latent semantic analysis (1998)
10. Frege, G.: Über sinn und bedeutung. Zeitschrift für Philosophie und philosophische Kritik 100, 25–50, translated, as 'On Sense and Reference', by Max Black
11. Grefenstette, E., Sadrzadeh, M.: Experimental support for a categorical compositional distributional model of meaning. CoRR abs/1106.4058 (2011)
12. Harris, Z.S.: Mathematical Structures of Language. Wiley, New York, NY, USA (1968)
13. Landauer, T.K., Dutnais, S.T.: A solution to platos problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. Psychological review pp. 211–240 (1997)
14. Lin, D.: Automatic retrieval and clustering of similar word. In: Proceedings of COLING-ACL. Montreal, Canada (1998)
15. Mitchell, J., Lapata, M.: Vector-based models of semantic composition. In: In Proceedings of ACL-08: HLT. pp. 236–244 (2008)
16. Mitchell, J., Lapata, M.: Composition in distributional models of semantics. Cognitive Science 34(8), 1388–1429 (2010)
17. Montague, R.: Formal Philosophy: Selected Papers of Richard Montague. Yale University Press (1974)
18. Pantel, P., Lin, D.: Document clustering with committees. In: SIGIR-02. pp. 199–206 (2002)
19. Pennacchiotti, M., Cao, D.D., Basili, R., Croce, D., Roth, M.: Automatic induction of framenet lexical units. In: EMNLP. pp. 457–465 (2008)
20. Salton, G., Wong, A., Yang, C.: A vector space model for automatic indexing. Communications of the ACM 18, 613–620 (1975)
21. Schütze, H.: Word space. In: Hanson, S.J., Cowan, J.D., Giles, C.L. (eds.) NIPS 5, pp. 895–902. Morgan Kaufmann Publishers, San Mateo CA (1993)
22. Schütze, H.: Automatic Word Sense Discrimination. Computational Linguistics 24, 97–124 (1998)
23. Turney, P.D., Pantel, P.: From frequency to meaning: Vector space models of semantics. Journal of artificial intelligence research 37, 141 (2010), doi:10.1613/jair.2934

QuestionCube: a framework for Question Answering

Piero Molino and Pierpaolo Basile

QuestionCube s.r.l.
{piero.molino,pierpaolo.basile}@questioncube.com
www.questioncube.com

Abstract. QuestionCube is a framework for Question Answering (QA) that combines several techniques to retrieve passages containing the exact answers for natural language questions. It exploits: (a) Natural Language Processing algorithms for question and candidate answers analysis both in English and Italian; (b) Information Retrieval probabilistic models for candidate answers retrieval and (c) Machine Learning methods for question classification. The data source for the answer is an unstructured text document collection stored in search indices. In this paper an overview of the QuestionCube framework architecture is provided, together with a description of Wikiedi, a QA system for Wikipedia which exploits the proposed framework.

1 Introduction

Question Answering (QA) emerged in the last decade as one of the most promising fields in Artificial Intelligence due to some competitions organized during international conferences [31, 25], but the first studies can be dated back to 1960s [3, 29]. In the last years some enterprise applications shown the potential of the state of the art technology, for example the IBM's Watson/DeepQA system [12, 11]. By exploiting techniques borrowed from Information Retrieval and Natural Language Processing (NLP), QA systems are able to answer user questions expressed in natural language with short passages of text which contain the exact answer or sometimes directly with the exact answer, depending on the domain, rather than returning long lists of full-text documents that users have to check in order to find the information needed, as most search engines do.

Most closed-domain QA systems use a variety of NLP methods to help the understanding of user's queries and the matching of passages extracted from documents [13, 15, 7]. The most commonly adopted linguistic analysis steps include: stemming, lemmatization with dictionaries, part-of-speech tagging, parsing, named entity recognition, lexical semantics (Word Sense Disambiguation), etc. The use of those NLP steps is fundamental to find the correct answer in closed-domain QA, since there is likely to be few answers to any user's question and the way in which they are expressed may be significatively different from the question. The difficulty of the task lies in mapping questions to answers by

way of uncovering complex lexical, syntactic, or semantic relationships between questions and candidate answers.

Open-domain QA systems, instead, have to face different types of problems: the probability of finding correct answers is higher, but the noise produced from the Web is also much higher than in the case of closed domain. Most systems exploit redundancy and textual pattern extraction and matching to solve the problem [9, 14, 24, 19].

The main limitation of current systems working on specific document collections is that they focus on precise tasks and are not general enough. On the other hand, open-domain systems, particularly those working on the World Wide Web, have long response times and lack in accuracy.

This paper describes QuestionCube, a framework for building QA systems with focus on closed domains, but which could be easily applied to open domains as well. It exploits NLP algorithms for both English and Italian and integrates a question categorization component based on Machine Learning techniques and linguistic rules written by human experts. Text document collections used as data sources are organized in indices for generic unstructured data storage with fast and reliable search functions exploiting state-of-the-art Information Retrieval weighting schemes.

The paper is structured as follows. Section 2 provides a generic overview of the framework architecture, while in Section 3 details about main components for analysis, search and filtering are described. Section 4 presents Wikiedi, a proof-of-concept system which relies on the QuestionCube framework and exploits Wikipedia pages as data source. Final conclusions, then, close the paper.

2 Framework overview

QuestionCube is a multilingual QA framework built using NLP and IR techniques.

The architecture, shown in Figure 1, is similar to the one proposed in [27], but it differs in several important aspects that make it more general and easier to expand. The first step is a linguistic analysis of the user's question. Question analysis is performed by a pipeline of NLP analyzer. The NLP components tag the question at different linguistic levels. The linguistic tagging process allows to classify the question according to a shared question-type hierarchy. The question classifier uses an ensemble learning approach that exploits both hand-written rules and rules inferred by machine learning categorization techniques, thus bringing together the hand-written rules' effectiveness and precision and the machine learning classifier's recall. The question is then passed to the search engines, whose architecture is highly parallel and distributed. Moreover, each single engine has its own query generator, because query's structure and syntax may change across different engines. The filter pipeline is then responsible for the scoring and the filtering of the passages retrieved by the search engines. Finally, the ranked list of passages is presented to the user.

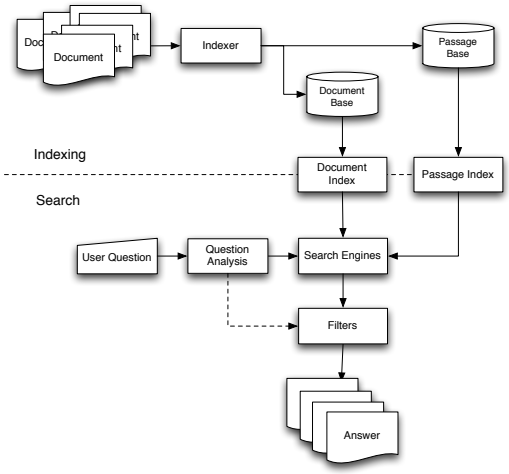


Fig. 1. QuestionCube architecture overview

The main motivation behind QuestionCube architecture is to create a Question Answering system simply by the dynamic composition of framework components. The high level of abstraction of the components allows to add support to a new language by just creating new interchangeable analyzers which implement the algorithms for the specific language. Another point that must be underlined is that our approach relies on several search engines in order to exploit different data source. For example, documents and passages could be retrieved from a database, from a Web search engine or from an enterprise search engine. The parallel approach allows to query several data sources at the same time.

3 Details

3.1 Question Analysis

The macro-component of the question analysis is composed of a pipeline of NLP analyzers, a data-structure to represent linguistic annotated text and the question classifier, as shown in Figure 2.

The NLP pipeline is easily configurable depending on the application domain of the QA system. Obviously, a small number of basic NLP analyzers added to the pipeline allows faster tagging, while more components in the pipeline requires more time for deeper linguistic analysis.

NLP analyzers are provided for both English and Italian. The stemmer is implemented by Snowball[1] both for English and Italian. The lemmatization is realized exploiting the morpho-syntactic analyzer of the WordNet API [10] for

[1] Available on-line: http://snowball.tartarus.org/

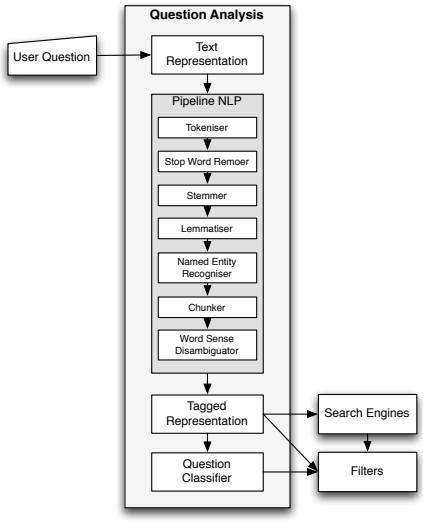


Fig. 2. Question analysis macro-component

the English, while Morph-it [32] is exploited for the Italian. Named Entity Recognition (NER) is performed by a machine learning classifier based on Support Vector Machines [8] using an open-source tool called YAMCHA [16]. The same tool is used for the chunker component. Both in chunking and NER, POS-tags and lemmas are adopted as features. The Word Sense Disambiguation (WSD) is implemented by the UKB algorithm [1], which is a graph-based technique based on a personalized version of PageRank [4] over WordNet graph.

The output of the NLP analyzers is a set of tags that are added to the text representation. The text representation is the input for the search engines, for the classifier and also for the filters, as they need linguistic information about the question to match it with the answers.

The NLP pipeline is also used by each filter to analyze the candidate answer at the same linguistic level as the question.

3.2 Question Classifier

The annotated text representation of the question is used by the question classifier. It is composed by three classifiers as shown in Figure 3.

The first one is based on Support Vector Machines and uses the tags from the text representation as features to classify the question. The main features are the head word of the question, the terms, their PoS tags, semantic identifiers provided by WSD and Named Entities.

The other two classifiers are rule-based ones that exploit respectively hand-written and learned rules in the form of regular expressions based on Named Entity categories and semantic identifiers.

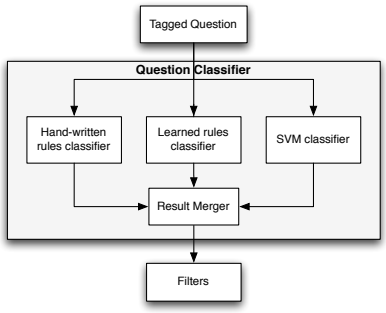


Fig. 3. Question classifier macro-component

The outputs of the classifiers are merged by using a weighted voting system that returns a question category.

The category is selected among the ones in the typology proposed in [17, 18]. Categories are exploited by filters in order to give a higher score to those candidate answers containing Named Entities in accordance with the question category.

3.3 Search Engine

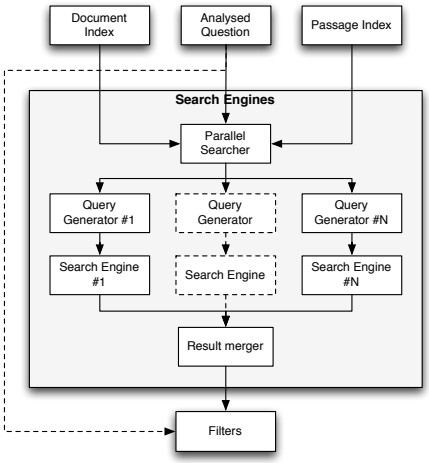


Fig. 4. Search engine macro-component

The search engine macro-component is designed to work in parallel and distributed environment. It allows to implement several information retrieval strategies and thus to aggregate their results, as shown in Figure 4.

The parallel engine is modular and it is possible to add an arbitrary number of different search engines inside it. It calls each engine when a new question comes and merges their outputs in a single list. The list contains all the candidate answers from all the engines, each one with a reference to the engines that retrieved it and the score assigned by each engine. Some filters normalize those scores in order to get an overall best score. Each single search engine has its own query generation component, because the syntax of the query may change among different engines. Each query generator may use different annotations from the text representation: some may use only tokens, others can use lemmas or stems, others may use WordNet synsets to generate the query. This approach allows to add a new search engine inside the framework with minimal effort. The main goal of using more than one search engine is to rely on different retrieval strategies in order to take the best results from each one. For example, in the current implementation, we adopt two search engines: the first one works on keywords, while the second one relies on lemmas. Moreover, the use of multiple search engines allows to use different retrieval models merging the results in an unique result set.

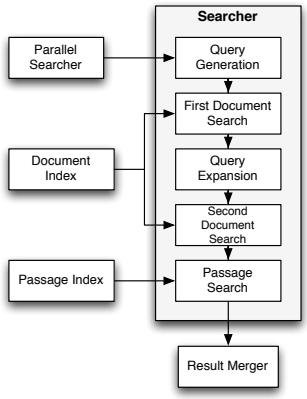


Fig. 5. Single search engine

The process performed by each search engine is described in Figure 5. The query generator builds the query for its search engine from the text representation provided by the parallel engine. Each query generator may implement different query improvement techniques (such as relevance feedback and query expansion). The query is executed by the search engine that returns the best scoring documents. The passage index is used to obtain the passages from retrieved

documents. These passages are merged into one single list by an aggregation component and then passed to the filters which score, sort and filter them.

The QuestionCube framework provides a search engine based on *BM25* model [26]. The query generation component for this searcher allows three different query improvement techniques:

- *Query expansion* through WordNet synonyms of the synsets found in the question;
- *Kullback-Liebler Divergence*, a statistical technique that exploits the terms distribution of the top-ranked documents [6, 20];
- *Divergence From Randomness*, a statistical technique that weights the terms distribution with the *Bo1* weighting scheme [2].

It is important to underline that the WordNet based query expansion is used only if the question has been disambiguated.

3.4 Filters

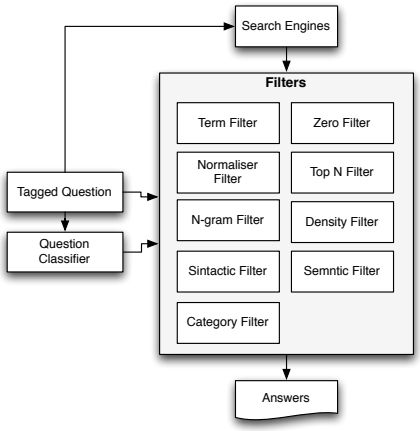


Fig. 6. Candidate answers filtering macro-component

This macro-component, sketched in Figure 6, contains all the passages filters. It allows to build a pipeline in which it is possible to add filters. If there is no dependence between the filters, it is possible to place them in any order to create different pipelines for several domains and needs.

Each filter checks every passage in input obtained from the search engine and assigns a score to them depending on the implemented logic. Each filter can exploit information provided by the text representation and use the category tag assigned to the question by the classifier. Some filters do not assign scores

but just sort the passages according to some score or ranking threshold. The composition of the filters in the pipeline is important to determine the quality of the results returned by the system, its efficiency and the time taken to give an answer.

A description of the logic of each filter is given below:

- **Zero Filter**: removes from the list all those passages that, at the moment of the analysis, have a general score of 0;
- **Top-N Filter**: sorts passages in a decreasing order according to their current score and removes all those passages under the N-th position in the ranking (N is given as input to the filter);
- **Terms filter**: assigns a score to every analyzed passage based on the frequency of the question terms in the passage;
- **Normalization Filter**: assigns a score to each analyzed passage based on the passage length, by normalizing its overall score. Both a simple normalization filter (which considers only the number of terms and is generally called *Byte-size Normalization*) and a filter based on the *Pivoted Normalised Document Length* technique are implemented. Both techniques and their effectiveness are discussed in [21, 30];
- **N-grams Filter**: assigns a score to each analyzed passage based on the overlapping of n-grams between the question and the passage (n is given as input to the filter);
- **Density Filter**: assigns a score to each analyzed passage based on the distance of the question terms inside the passage increasing the score of those passages in which the question terms are closer. The density is calculated by the Minimal Span Weighting schema proposed by [22]:
  $\left(\frac{|q \cap d|}{1+\max(mms)-\min(mms)}\right)^{\alpha} \left(\frac{|q \cap d|}{|q|}\right)^{\beta}$
  where q and d are the set of terms respectively of the query and the document (specifically here, the query is the question and the document is the passage); $\max(mms)$ and $\min(mms)$ are the initial and final location of the sequence of document terms containing all the query terms; and α and β are two parameters.
- **Syntactical Filter**: assigns a score to each analyzed passage based on the Phrase Matching algorithm, presented in [23]. The algorithm takes into account the head of each phrase. If the head is common to the two considered texts (in this case the query and the passage), the maximal overlapping length of each phrase is calculated.
- **Semantic Filter**: assigns a score to each analyzed passage based on the frequency of terms tagged with the same WordNet synsets inside both question and passage. A more complex filter that calculates a semantic similarity measure between texts based on the semantic distance measure described in [5] is one of the future developments;
- **Category Filter**: assigns a score to each analyzed passage based on a list of pairs that link the question categories to typologies of named entity: if, on the basis of the question category, entities of the expected typology are found in the passage the score will be positive.

– **Z-Score Filter**: assigns a score to each analyzed passage based on the Z-Score normalization [28] of scores assigned by search engines and other filters.

A boost factor can be assigned to each filter which intensifies or decreases its strength.

4 Wikiedi

Wikiedi is a Web application that allows users to ask questions and receive answers extracted from articles from Italian and English Wikipedia. The Question Answering core of Wikiedi is built on the QuestionCube framework with a specific configuration that balances accuracy and reactivity.

The system is configured to index Wikipedia pages with their respective linguistic annotations. This ensures quick response time because NLP algorithms will not process linguistically each passage at runtime. To improve performances, the annotated passages are represented in a compact binary structure stored in a database. This allows fast passage retrieval reducing to zero the reconstruction time.

The filters adopted in Wikiedi range from the most basic ones that work on tokens to the most sophisticated ones exploiting semantics.

The decision to use documents from Wikipedia to evaluate the potential of QuestionCube framework is motivated by the heterogeneous nature of the information on Wikipedia. This reflects the enterprise context where documents that belong to different domains are stored in a single collection increasing the noise in the retrieval phase.

The other goal of Wikiedi is to engage the user in improving system performances. After the user has submitted a question to the system, Wikiedi will display an ordered list of answers. The user will have the possibility of voting for the correct answer, so that the system can use the feedback to improve precision and recall in future queries. The next time the question is issued, the results will be sorted by mixing the score given by the system and users' judgements.

Moreover, when one of the answers provided by Wikiedi is not correct, users will have the opportunity of inserting the correct one. Using this strategy it is possible to enrich the system with additional information. Users asking the same question will then obtain both automatically obtained results alongside with user added answers.

To meet users information needs, the QuestionCube framework also allows to implement "similar questions" function easily by indexing user questions as they are asked and calculating their similarity. Moreover, the framework allows to implement a simple content-based recommender system that suggests questions the user may be also interested in.

The results are shown segment by segment, as shown in Figure 7. Clicking on a result, a page of the full Wikipedia article text is shown. The page is automatically enriched mashing up several multimedia contents from Web 2.0 websites such as Fotopedia, Flickr, Youtube and Vimeo.

Fig. 7. Wikiedi web interface

The Italian version of Wikiedi is available on-line: www.wikiedi.it. The English version will follow soon on www.wikiedi.com.

As for the evaluation, currently statistics about Wikiedi performances are currently not available, since a large number of users' feedback is needed to evaluate them. However, an evaluation of a system built with the QuestionCube framework has been performed using a standard dataset adopted in QA called CLEF 2010 ResPubliQA [25] based on multi-lingual documents from European Legislation. The dataset consists of 10,855 documents and 200 questions. The system is evaluated using the c@1 measure, which takes into account the accuracy on the first returned passage. Table 1 reports the results of our system for each language. The last column shows the results obtained by the best participant system. The obtained results show improvements both in English and Italian.

Table 1. Results on CLEF 2010 ResPubliQA with c@1 measure

Language	QuestionCube	Best system
Italian	**0.68**	0.63
English	**0.75**	0.73

5 Conclusions

In this paper, the QuestionCube framework has been presented. QuestionCube finds the correct answer to a question by combining Natural Language Processing algorithms, Information Retrieval probabilistic models and Machine Learn-

ing methods. Wikiedi was also presented as an example enterprise application. Wikiedi allows the user to ask questions in natural language on Wikipedia pages combining the power of the QuestionCube framework with feedback and additional information provided by the community of users. Finally, an evaluation on a standard dataset, CLEF 2010 ResPubliQA, has been provided, which shows an improvement in comparison to other state-of-the-art systems.

References

1. Agirre, E., Soroa, A.: Personalizing PageRank for word sense disambiguation. In: Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics. pp. 33–41. EACL '09, Association for Computational Linguistics, Stroudsburg, PA, USA (2009)
2. Amati, G., Van Rijsbergen, C.J.: Probabilistic models of information retrieval based on measuring the divergence from randomness. ACM Trans. Inf. Syst. 20, 357–389 (October 2002)
3. Bert F. Green, J., Wolf, A.K., Chomsky, C., Laughery, K.: Baseball, an automatic question-answerer. Managing Requirements Knowledge, International Workshop on 0, 219 (1961)
4. Brin, S., Page, L.: The Anatomy of a Large-Scale Hypertextual Web Search Engine. In: Seventh International World-Wide Web Conference (WWW 1998) (1998)
5. Budanitsky, A., Hirst, G.: Evaluating WordNet-based Measures of Lexical Semantic Relatedness. Comput. Linguist. 32, 13–47 (March 2006)
6. Carpineto, C., de Mori, R., Romano, G., Bigi, B.: An information-theoretic approach to automatic query expansion. ACM Trans. Inf. Syst. 19, 1–27 (2001)
7. Chen, J., Diekema, A., Taffet, M.D., McCracken, N.J., Ozgencil, N.E., Yilmazel, O., Liddy, E.D.: Question answering: Cnlp at the trec-10 question answering track. In: TREC (2001)
8. Cortes, C., Vapnik, V.: Support-vector networks. Machine learning 20(3), 273–297 (1995)
9. Dumais, S., Banko, M., Brill, E., Lin, J., Ng, A.: Web question answering: is more always better? In: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval. pp. 291–298. SIGIR '02, ACM, New York, NY, USA (2002)
10. Fellbaum, C.: WordNet: an electronic lexical database. Language, speech, and communication, MIT Press
11. Ferrucci, D.A.: Ibm's watson/deepqa. SIGARCH Computer Architecture News 39(3) (2011)
12. Ferrucci, D.A., Brown, E.W., Chu-Carroll, J., Fan, J., Gondek, D., Kalyanpur, A., Lally, A., Murdock, J.W., Nyberg, E., Prager, J.M., Schlaefer, N., Welty, C.A.: Building Watson: An Overview of the DeepQA Project. AI Magazine 31(3), 59–79 (2010)
13. Harabagiu, S.M., Moldovan, D.I., Pasca, M., Mihalcea, R., Surdeanu, M., Bunescu, R.C., Girju, R., Rus, V., Morarescu, P.: Falcon: Boosting knowledge for answer engines. In: TREC (2000)
14. Harabagiu, S.M., Paşca, M.A., Maiorano, S.J.: Experiments with open-domain textual question answering. In: Proceedings of the 18th conference on Computational linguistics - Volume 1. pp. 292–298. COLING '00, Association for Computational Linguistics, Stroudsburg, PA, USA (2000)

15. Hovy, E.H., Gerber, L., Hermjakob, U., Junk, M., Lin, C.Y.: Question answering in webclopedia. In: TREC (2000)
16. Kudo, T., Matsumoto, Y.: Fast Methods for Kernel-Based Text Analysis. In: Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics. pp. 24–31. ACL, Sapporo, Japan (July 2003)
17. Li, X., Roth, D.: Learning question classifiers. In: Proceedings of the 19th international conference on Computational linguistics - Volume 1. pp. 1–7. COLING '02, Association for Computational Linguistics, Stroudsburg, PA, USA (2002)
18. Li, X., Roth, D.: Learning question classifiers: the role of semantic information. Nat. Lang. Eng. 12, 229–249 (September 2006)
19. Lin, J.: An exploration of the principles underlying redundancy-based factoid question answering. ACM Trans. Inf. Syst. 25 (April 2007)
20. Lv, Y., Zhai, C.: Positional relevance model for pseudo-relevance feedback. In: Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval. pp. 579–586. SIGIR '10, ACM, New York, NY, USA (2010)
21. Manning, C.D., Raghavan, P., Schtze, H.: Introduction to Information Retrieval. Cambridge University Press, New York, NY, USA (2008)
22. Monz, C.: Minimal span weighting retrieval for question answering. In: Gaizauskas, R., Greenwood, M., Hepple, M. (eds.) Proceedings of the SIGIR Workshop on Information Retrieval for Question Answering. pp. 23–30 (2004)
23. Monz, C., de Rijke, M.: Tequesta: The university of amsterdam's textual question answering system. In: TREC (2001)
24. Paşca, M.: Open-domain question answering from large text collections. Studies in computational linguistics, CSLI Publications (2003)
25. Penas, A., Forner, P., Rodrigo, A., Sutcliffe, R.F.E., Forascu, C., Mota, C.: Overview of ResPubliQA 2010: Question Answering Evaluation over European Legislation. In: Braschler, M., Harman, D., Pianta, E. (eds.) Working notes of ResPubliQA 2010 Lab at CLEF 2010 (2010)
26. Robertson, S., Zaragoza, H.: The probabilistic relevance framework: Bm25 and beyond. Found. Trends Inf. Retr. 3, 333–389 (April 2009)
27. Schlaefer, N., Gieselman, P., Sautter, G.: The ephyra qa system at trec 2006. In: TREC (2006)
28. Shaw, J.A., Fox, E.A., Shaw, J.A., Fox, E.A.: Combination of multiple searches. In: The Second Text REtrieval Conference (TREC-2. pp. 243–252 (1994)
29. Simmons, R.: Answering english questions by computer: A survey. Communications of the ACM 8(1), 53–70 (1965)
30. Singhal, A., Buckley, C., Mitra, M.: Pivoted document length normalization. In: Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval. pp. 21–29. SIGIR '96, ACM, New York, NY, USA (1996)
31. Voorhees, E.M., Tice, D.M.: The trec-8 question answering track evaluation. In: In Text Retrieval Conference TREC-8. pp. 83–105 (1999)
32. Zanchetta, E., Baroni, M.: Morph-it! a free corpus-based morphological resource for the italian language. Corpus Linguistics 2005 1(1) (2005)

TV-Show Retrieval and Classification

Cataldo Musto[1], Fedelucio Narducci[1], Pasquale Lops[1],
Giovanni Semeraro[1], Marco de Gemmis[1],
Mauro Barbieri[2], Jan Korst[2], Verus Pronk[2], and Ramon Clout[2]

[1] Department of Computer Science, University of Bari "A. Moro", Italy,
{cataldomusto,narducci,lops,semeraro,degemmis}@di.uniba.it
[2] Philips Research, Eindhoven, The Netherlands,
{mauro.barbieri,jan.korst,verus.pronk,ramon.clout}@philips.com

Abstract. Recommender systems are popular tools to aid users in finding interesting and relevant TV shows and other digital video assets, based on implicitly defined user preferences. In this context, a common assumption is that user preferences can be specified by program types (such as documentary, sports), and that an asset can be labeled by one or more program types, thus allowing an initial coarse preselection of potentially interesting assets. Furthermore each asset has a short textual description, which allows us to investigate whether it is possible to automatically label assets with program type labels. We compare the Vector Space Model (VSM) with more recent approaches to text classification, such as Logistic Regression (LR) and Random Indexing (RI) on a large collection of TV-show descriptions. The experimental results show that LR is the best approach, but RI outperforms VSM under particular conditions.

Keywords: Vector Space Model, Random Indexing, Logistic Regression

1 Introduction

Automatic TV recommendations have been explored extensively in the literature where most papers assume that the set of items for recommendations is of moderate size. Most approaches are not directly applicable to web video repositories (such as YouTube) whose item sets are orders of magnitude larger. To provide personalized recommendations for digital assets on the web and TV, a possible approach is to match the assets' textual descriptions to personal preferences of users. It is common practice to classify TV shows by labeling them with one or more *program type* labels. It may also be assumed that user preferences can be coarsely expressed in terms of program types [2]. In this paper, we assume that each asset has a short textual description and we investigate (a) how well that description can be automatically mapped to a program type and (b) which machine learning algorithms are best suited for the above mentioned classification task. To this end, we have extensively tested algorithms using a large collection of TV-show descriptions which calls for the adoption of simple and scalable retrieval models. A text classification algorithm based on the Vector Space Model

(VSM) might be a good solution, provided that effective dimensionality reduction techniques are integrated, such as Random Indexing (RI) [3]. As regards classification algorithms, we opted for Logistic Regression (LR), since it is generally considered as accurate as Support Vector Machines, with the advantage of yielding a probability model [4].

This research is carried out in the context of a joint project with APRICO Solutions[3], a software company and part of Philips Electronics. APRICO Solutions develops video recommender and targeting technology, primarily for the broadcast and internet industries. Further details are available in [1].

2 TV-show Classification and Retrieval

The two problems we focus upon can be defined as follows:
TV-show classification: given a program description s and a set P of program types, choose a program type $p \in P$ that best matches the program description. Each TV show has exactly one label assigned to it.
TV-show retrieval: given a set S of TV-show descriptions and a program type $p \in P$, return a ranked list of k TV-show descriptions from S that best match program type p.

Three approaches for the TV-show classification and TV-show retrieval tasks have been investigated. We compare VSM with LR and RI. For both tasks, TV-show textual descriptions have been preprocessed for obtaining bag-of-words representations (BOW).

2.1 TV-SHOW CLASSIFICATION

Vector Space Model Given a set of documents (*corpus*), each document is represented as a point in a *n-dimensional* vector space (n is the cardinality of the vocabulary). Formally, each document is represented as a vector $\boldsymbol{d} = (w_1, \ldots, w_n)$ where w_i is the TFIDF score of the feature i. A vector space representation of each program type is obtained by summing the vectors of TV shows belonging to that program type. Thus, given a TV show s to be classified, its program type is given by the program type vector with the highest cosine similarity to s. VSM has some important limitations: it is not incremental and it does not model semantics.
Random Indexing. RI is a scalable and incremental dimensionality reduction technique. It belongs to the class of *distributional models*, which state that the meaning of a word can be inferred by analyzing its use (*distribution*) within a corpus of textual data. Random Indexing for TV-show classification follows the same steps as for VSM: a prototype vector is built for each program type and the cosine similarity between a TV-show and each program type is computed. Unlike VSM, these steps are performed on the reduced vector space obtained as output of the RI algorithm (500, 700 dimensions).

[3] `www.aprico.tv`

Logistic Regression. LR is a supervised learning algorithm based on a generalized linear model. In this work we exploited the implementation provided in LIBLINEAR[4]. Given a TV show, we compute the probability of each program type by exploiting the logistic functions learned for each class. The TV-show program type is determined by the highest probability.

2.2 TV-SHOW RETRIEVAL

For the TV-show retrieval task, we exploited only LR and RI, since they achieved the best performance for most classes in the classification task.
Random Indexing. As in the classification task, the vector space is reduced through the RI algorithm. Given a prototype vector built for each program type, the cosine similarity with all TV shows is computed in order to get the list of the best matching TV-show descriptions for a specific program type.
Logistic Regression. The probability that a TV show belongs to a specific program type is computed for the retrieval task as well. In this task, given a program type p, the TV shows are ranked based on their probability to belong to p and are returned in a ranked list.

Program Type	VSM	RI		LR
		500	700	
miscellaneous	0.11	**0.37**	**0.35**	**0.26**
movies	0.76	0.35	0.40	**0.83**
short movies	0.35	**0.95**	**0.95**	**0.75**
tv series	0.74	0.47	0.58	**0.87**
sport	0.90	**0.90**	**0.91**	**0.96**
show	0.65	0.48	0.48	**0.85**
events	0.63	**0.72**	**0.74**	**0.86**
documentary	0.63	0.23	0.24	**0.72**
reportage	0.57	0.41	0.43	**0.75**
report	0.15	**0.32**	**0.30**	**0.43**
magazine	0.64	0.39	0.36	**0.81**
news	0.54	**0.84**	**0.83**	**0.82**
videoclip	0.79	**0.94**	**0.92**	**0.83**
advertising	0.94	**0.99**	**0.99**	**0.98**
music	0.81	**0.83**	**0.81**	**0.84**

Fig. 1. Accuracy of VSM, RI, and LR for the classification task.

Alg	Dim	Precision@n%					
		5%	10%	25%	50%	75%	100%
RI	500	0.58	0.52	0.46	0.41	0.38	0.36
RI	1000	0.58	0.53	0.47	0.42	0.38	0.36
RI	1500	0.57	0.53	0.47	0.41	0.39	0.35
RI	2000	0.57	0.53	0.46	0.41	0.34	0.35
LR		**0.92**	**0.90**	**0.88**	**0.86**	**0.82**	**0.75**

Fig. 2. $P@n\%$ of RI, and LR for the retrieval task.

3 Experimental Evaluation

The goal of the experimental evaluation is to measure the effectiveness of the VSM, RI, and LR models in the retrieval and classification tasks. The experiment

[4] www.csie.ntu.edu.tw/~cjlin/liblinear/

has been carried out through a *k-fold cross validation* (k=10), on a dataset composed of 133,579 TV shows broadcast from a set of 47 channels in the German language. The textual descriptions are the input to the learning process and are represented by bag of words. Stemming and stop-words elimination are performed on the text. For the *classification* task we used the Accuracy as metric: it is calculated as the ratio between the TV shows correctly classified and the total number of TV shows classified. For the *retrieval* task we used the Precision@n%: it is calculated as the ratio between the TV shows correctly classified and the n% of the Test Set. VSM, LR, and RI (using different vector space dimensions) have been compared.

Classification task. Figure 1 reports accuracy values of VSM, LR and RI. The configurations that overcome the baseline (VSM) are in bold. For some classes the dimensionality reduction technique deteriorated the performance of the classifier. However for most classes, RI outperformed VSM, even though the reduction of the vector space dimension is considerable. Furthermore, the LR algorithm obtained the best accuracy. The best improvement achieved compared to the VSM model is almost 20%.

Retrieval task. In general the different space dimensions for random indexing do not affect the retrieval accuracy of the retrieval model (see Figure 2). Also for this task LR achieved better results compared to RI. The accuracy of the model decreases when the size of the retrieved list increases. This was expected because less relevant shows for each program type are in the tail of the list.

4 Conclusions and Future Work

The best performing approach for the classification task was LR. Despite the fact that this approach already showed to be effective in text classification in the literature, results achieved in this specific scenario were not obvious, since TV shows have very short textual descriptions and only few training examples were available for many classes. RI demonstrated a good performance in TV-show classification for the classes with a small number of instances in the training set. In the retrieval task LR outperforms the other approaches as well. In the future we will work in a recommendation scenario in order to re-rank the retrieved list of TV shows according to the user preferences.

References

1. C. Musto and F. Narducci. Tv-show retrieval and classification. Technical report, Philips Research, High Tech Campus, Eindhoven, The Netherlands, July 2011.
2. V. Pronk, J. Korst, M. Barbieri, and A. Proidl. Personal television channels: simply zapping through your pvr content. In *Proceedings of the 1st International Workshop on Recommendation-based Industrial Applications*, RecSys '09, 2009.
3. M. Sahlgren. An introduction to random indexing. In *Methods and Applications of Semantic Indexing Workshop, TKE 2005*, 2005.
4. T. Zhang and F. J. Oles. Text categorization based on regularized linear classification methods. *Information Retrieval*, 4:5–31, 2000.

Un prototipo per la ricerca di opinioni sui blog dedicati alle trasmissioni televisive d'interesse nazionale

Giambattista Amati, Marco Bianchi, and Giuseppe Marcone

Fondazione Ugo Bordoni
Viale del Policlinico, 147
00161 Rome, Italy
gba@fub.it
mbianchi@fub.it
gmarcone@fub.it

Sommario In questo lavoro si riporta l'esperienza maturata durante la realizzazione di un prototipo per la ricerca delle opinioni pubblicate sui blog dedicati ai programmi televisivi trasmessi dalle emittenti italiane. Il contributo per la comunità scientifica italiana dell'Information Retrieval è duplice: da un lato si presenta il primo benchmark per il task dell'opinion finding applicato a piattaforme di blog in lingua italiana e si riporta la metodologia adottata per la sua creazione. In secondo luogo si descrive l'architettura di un sistema che implementa un algoritmo dictionary-based di comprovata efficacia utile ad affrontare il problema dell'opinion finding su testi in lingua italiana. Tale sistema, basato su componenti open-source, supporta la creazione di ulteriori benchmark a partire dai quali genera in modo automatico i dizionari necessari al funzionamento dell'algoritmo che implementa. Proprio quest'ultima funzionalità è da considerarsi strategica per la comunità scientifica vista la scarsa disponibilità di risorse linguistiche italiane e il costo necessario alla loro creazione e aggiornamento.

Keywords: Information Retrieval, Sentiment Analysis, Opinion Finding

1 Introduzione

La disciplina scientifica il cui fine è lo sviluppo di tecniche di estrazione della conoscenza da documenti contenenti opinioni è nota in letteratura con il nome di *sentiment analysis*. Oggigiorno le principali comunità scientifiche che si occupano di sentiment analysis sono due: la comunità dell'Intelligenza Artificiale, che utilizza prevalentemente tecniche di Processamento del Linguaggio Naturale (NLP) finalizzate alla classificazione automatica dei documenti e di estrazione puntuale di informazioni da documenti [10], e la comunità dell'Information Retrieval, che ha specializzato il problema al mondo del Web. La differenza fondamentale tra le ricerche effettuate dalla due comunità risiede nella tipologia di collezioni che prendono a riferimento.

Infatti la comunità dell'Intelligenza Artificiale basa i sui studi, nella maggior parte dei casi, su collezioni composte esclusivamente da documenti contenenti opinioni e strutturalmente omogenei. In questi casi il problema diventa quello di classificare, ad

esempio, i documenti contenenti opinioni positive da quelli contenenti opinioni negative, oppure quello di estrarre informazioni puntuali, come le caratteristiche più o meno apprezzate di un prodotto commerciale [4,11].

Diversamente numerosi studi della comunità dell'Information Retrieval sono basati su collezioni Web. Tali collezioni sono caratterizzate, tra l'altro, dalla presenza di documenti non contenti opinioni e dalla eterogeneità a livello strutturale delle pagine Web, quasi sempre scaricate da siti diversi. In questo scenario la sentiment analysis viene generalmente considerato un problema di re-rank a due fasi [7]: nella prima si cerca di individuare i documenti che sono rilevanti rispetto all'esigenza informativa dell'utente (topic), indipendentemente dalla presenza di opinioni rispetto al topic cercato; nella seconda l'insieme dei documenti individuati a valle di un processo di riordinato (re-rank) in funzione presenza o assenza di opinioni. L'intero processo di recupero, denominato *opinion-finding*[1], ha quindi l'obiettivo di recupere pagine Web contenenti opinioni rispetto ad un determinato topic.

Questo lavoro si inquadra nell'ambito delle attività finalizzate alla realizzazione del prototipo di un motore ricerca in grado di trovare le opinioni che i telespettatori di programmi televisivi riportano sui blog in lingua italiana. Tale motore può essere utile sia ai telespettatori che vogliono leggere, o scrivere, recensioni o commenti relativi ai loro programmi preferiti, sia alle emittenti televisive che intendono indagare l'opinione del popolo del Web in merito ai programmi trasmessi.

A partire dall'analisi dei requisi è stato eseguito uno studio dello stato dell'arte che ha confermato quanto già riportato in [7] e cioè che le principali tecniche di opinion-finding possono essere classificate in due principali categorie: strategie basate su classificatori (classification-based) e strategie basate su dizionari (lexicon-based). Considerata l'efficacia dimostrata da quest'ultima classe di tecniche, si è deciso di applicare la strategia lexicon-based presentata in [1]. Tale tecnica è di particolare interesse non solo perchè si è dimostrata tra le più performanti nelle varie edizioni della TREC, ma anche perchè permette la creazione *automatica* di dizionari di termini "portatori" di opinione (opinion-bearing terms). Considerato che non esistono, ad oggi, dizionari italiani per la sentiment analysis si ritiene che la realizzazione di un prototipo che supporti la generazione automatica di dizionari italiani sia da considerarsi un valore aggiunto per l'intera comunità scientifica italiana. Il basso costo di generazione, e quindi di aggiornamento, del dizionario va infatti incontro a quel requisito di economicità che dovrebbe contraddistinguere la voce "costo di manutenzione" di ogni sistema software. È tuttavia necessario precisare che a fronte di un risparmio in termini di impiego di risorse umane, si è costretti ad accettare la presenza, all'interno del dizionario, di termini "intrusi", ovvero termini che, almeno in apparenza, non sono portatori di opinione.

Il prototipo, presentato nella Sezione 4, è caratterizzato dall'originale integrazione tra la catena di tool nutch-solr [3,12], lo standard di fatto della comunità dell'open-source per la realizzazione di motori di ricerca, e il framework Terrier [8], strumento di Information Retrieval estremamente diffuso nella comunità scientifica e necessario per l'implementazione della tecnica presentata in [1]. Grazie a tale integrazione è possibile soddisfare anche due importanti requisiti non funzionali aventi come obiettivo la realizzazione di un motore di ricerca che sia:

[1] Nomenclatura introdotta nell'ambito della Blog Track di TREC 2006 [9,13].

1. in grado di scalare alle dimensioni tipiche del Web, proprio per rendere possibile il monitoraggio di una porzione significativa della blogosfera italiana;
2. caratterizzato da un basso costo di realizzazione e manutenzione.

Il prototipo supporta anche il processo per la realizzazione di generici benchmark per l'Information retrieval. Proprio grazie a tale supporto, è stato creato il primo benchmark per la sperimentazione di soluzioni al problema dell'opinion finding su testi italiani, come riportato nella Sezione 3.

Tra i contributi del lavoro si evidenzia la descrizione di due diverse strategie per l'acquisizione dei contenuti pubblicati su blog con relativi vantaggi e svantaggi. Le considerazioni relative ai due approcci, oggetto della Sezione 2, sono generalizzabili a tutti i contesti in cui i contenuti su cui effettuare le ricerche sono fortemente condizionati da eventi esterni alla Rete quali, appunto, la trasmissione di un programma televisivo o la diffusione di notizie.

Infine la Sezione 5 conclude il lavoro.

2 Metodologie per l'acquisizione dei contenuti di un blog

Al fine di rendere più chiara la presentazione delle metodologie per l'acquisizione dei contenuti pubblicati su un blog, le componenti logiche di una piattaforma di blogging da tenere in considerazione sono:

- *permalink*, o link permanente: URL relativo a una pagina Web che contiene un post e i relativi commenti. Il contenuto testuale raggiungibile con permalink è l'obiettivo finale dell'attività di acquisizione;
- *homepage*: pagina dinamica sulla quale vengono riportati gli ultimi post pubblicati e i relativi permalink;
- *navigatore*: strumento che implementa una tecnica di "navigazione a faccette" (faceted search) al fine di semplificare la ricerca dei post di interesse. In genere su tutte le pagine di un blog sono presenti un numero significativo di navigatori;
- *pagina di aggregazione*: pagine dinamiche che contengono tutti i post che soddisfano un criterio di navigazione a faccette (ad esempio tutti i post pubblicati in un determinato mese);
- *RSS feed*: file in formato RSS (Really Simple Syndication)[2] sul quale vengono periodicamente riportati i permalink degli ultimi post pubblicati.

A partire dalle componenti logiche appena elencate, l'acquisizione dei contenuti pubblicati su una piattaforma di blog può avvenire adottando due diverse strategie a seconda delle esigenze.

La prima strategia consiste nell'effettuare una sorta di "fotografia" dei contenuti presenti sull'intero blog mediante attività di *crawling*. In questo caso l'idea è quella di fornire al crawler la URL della homepage e lasciare a quest'ultimo la responsabilità di navigare (in modo automatico) sul blog al fine di scaricarne i contenuti. Il vantaggio di questa tecnica è dato dalla completezza del risultato (alta recall): ciò significa che al

[2] Per le specifiche del procollo RSS far riferimento al sito `http://www.rssboard.org/`

termine dell'attività di crawling tutti i contenuti indirizzati da permalink saranno stati scaricati. Lo svantaggio principale sarà dato dalla bassa precisione (precision) poichè un crawler non è in grado di distinguere un permalink da altre URL (a meno dello sviluppo di filtri di URL specializzati per le singole piattaforme di blog, operazione che però ha controindicazioni in termini di costo di scalabilità e manutenzione del sistema). Di conseguenza il crawler scaricherà anche l'homepage e, soprattutto, le pagine di aggregazione di post. Queste ultime sono da considerarsi "rumore", in quanto replicano il testo dei post già raggiungibile seguendo i permalink. Vale la pena evidenziare che il numero di pagine di aggregazione è proporzionale al numero di navigatori e che può, di conseguenza, anche essere consistente.

La seconda strategia consiste nell'individuare e scaricare i permalink dei nuovi post mediante il *monitoraggio* degli RSS feed. Questa tecnica, adottata per la realizzazione di due benchmark internazionali utilizzati nell'ambito delle gare TREC [6], ha il vantaggio di produrre una elenco composto esclusivamente da permalink. Purtroppo però il monitoraggio degli RSS non permette lo scaricamento dei vecchi permalink, ossia delle URL che sono già state eliminate dall'RSS perché la pubblicazione del post da loro riferito non rappresenta più una "novità" per gli utenti del blog.

Indipendentemete dai vantaggi e dagli svantaggi, l'adozione della prima strategia è obbligatoria quando si vuole includere nella collezione post poco recenti, o quando non si è nella condizione di aspettare il tempo necessario per eseguire il monitoraggio degli RSS. Nel caso dell'acquisizione dei contenuti pubblicati su blog che trattano trasmissioni televisive, la prima strategia è da considerarsi una scelta obbligata anche quando le trasmissioni di interesse sono già andate in onda.

3 Un benchmark per l'opinion finding task in lingua italiana

In genere un tipico benchmark di Information Retrieval è composto da:

1. *un insieme di topic*, ovvero un elenco di esigenze informative, esprimibili come query, definite da esperti di dominio in modo tale da essere rappresentative dell'utenza reale;
2. una *collezione di documenti*;
3. un *insieme di valutazioni*, ottenute grazie all'apporto degli esperti di dominio, nel quale a ogni topic viene associato un sottoinsieme di documenti della collezione rilevante rispetto a tale topic.

Coerentemente con quanto appena riportato, la creazione del benchmark per l'opinion finding task applicato al dominio delle trasmissioni televisive trasmesse da emittenti TV ha richiesto le seguenti attività:

- definizione di un elenco di trasmissioni televisive su cui eseguire ricerche (topics) e che, almeno sulla carta, suscitino dibattito tra gli utenti del Web. Nel caso specifico sono state individuate 65 trasmissioni televisive di vario genere (es. attualità, reality, fiction, satira, ecc.);
- individuazione delle piattaforme di blog dalle quali acquisire i contenuti: grazie al coinvolgimento di esperti di dominio è stato stilato un elenco di 100 URL (seeds) relative a piattaforme di blog tematiche;

– conduzione dell'attività di crawling. Considerato che molti dei programmi selezionati non andavano in onda durante il periodo dedicato all'acquisizione dei contenuti (periodo che va dai primi di novembre 2010 e alla prima metà di dicembre 2010), si è deciso di adottare la strategia del crawling dei blog. Al termine dell'attività di crawling la collezione risulta composta da 6.067.494 pagine HTML, tra le quali sono compresi permalink, homepage, pagine di aggregazione e duplicati (per lo più ottenuti a causa dell'utilizzo di pagine dinamiche da parte delle piattaforme di blog);
– rimozione dei duplicati. Successivamente alla fase di crawling, le pagine duplicate sono state rimosse a seguito di un controllo sul valore MD5 e riducendo il numero di documenti della collezione a 1.531.837 pagine Web;
– creazione dell'insieme delle valutazioni. Dopo aver indicizzato l'intera collezione con Lucene, sono state selezionate 30 trasmissioni televisive tra le 65 precedentemente individuate e, per ognuna di queste, è stato eseguito un recupero di 200 risultati utilizzando il nome della trasmissione come query e il search handler di seguito riportato[3]:

```
<requestHandler name="/topicSearch"
  class="solr.SearchHandler">
    <lst name="defaults">
      <str name="defType">dismax</str>
      <str name="echoParams">explicit</str>
      <float name="tie">0.01</float>
      <str name="qf">content^1.0</str>
      <str name="pf">anchor^1.0 title^0.1</str>
      <int name="ps">3</int>
      <str name="fl">url</str>
      <bool name="hl">false</bool>
    </lst>
</requestHandler>
```

Per ognuna delle 6.000 URL così individuate, un esperto di dominio ha registrato (mediante un'applicazione Web appositamente sviluppata) il proprio parere in merito a:

- la pertinenza della pagina recuperata rispetto al topic. Più precisamente la domanda di riferimento per i valutatori è stata: "La pagina Web associata alla URL è pertinente rispetto alla trasmissione televisiva in oggetto?" con possibili valori di risposta: *rilevante* e *non rilevante*.
- la tipologia di pagina Web. In questo caso la domanda di riferimento per i valutatori è stata: "La pagina Web associata alla URL è una home-page, una pagina di aggregazione o un permalink?", con possibili valori di risposta: *homepage*, *pagina di aggregazione post*, *permalink* o *altro*.
- la presenza di opinioni nella pagina Web, con la seguente domanda di riferimento: "La pagina Web associata alla URL contiene opinioni positive, opinioni negative, opinioni miste o nessuna opinione?" con possibili valori di risposta: *nessuna opinione*, *opinioni positive*, *opinioni negative* o *opinioni miste*.

[3] Per approfondimenti sui parametri del request handler si veda `http://wiki.apache.org/solr/DisMaxQParserPlugin`

4 Un prototipo per la ricerca di opinioni sui blog

Nell'ambito del progetto TV++ condotto dalla Fondazione Ugo Bordoni e dall'Istituto Superiore delle Comunicazioni e delle Tecnologie dell'Informazione è stato realizzato un prototipo per l'applicazione della tecnica dell'opinion finding al dominio dei blog dedicati ai programmi televisivi trasmessi dalle emittenti italiane.

Come già anticipato nella Sezione 1, il prototipo implementa la tecnica dictionary-based presentata in [1]. In estrema sintesi tale tecnica prevede due passi principali:

1. Costruzione automatica di un dizionario composto da termini che caratterizzano i documenti in cui vengono espresse opinioni (termini *opinion-bearing*). A ogni termine del dizionario è associato un peso che fornisce, informalmente parlando, una misura del suo grado di soggettività, ove per termine soggettivo s'intende un termine che usualmente compare in una frase soggettiva. Ad esempio i termini "credo" e "penso" si suppone che abbiano un grado di soggettività alto in quanto spesso usati in frasi che esprimono opinioni. La costruzione automatica del dizionario avviene adottando un approccio di tipo statistico-probabilistico basato su modelli della famiglia *Divergence from Randomness* (DFR) [2].
2. Esecuzione di un algoritmo di opinion retrieval che, sfruttando le informazioni presenti nel dizionario appena descritto, assegni ad ogni documento uno score funzione sia della rilevanza rispetto alla query, sia della presenza di opinioni nel testo. L'algoritmo tenderà pertanto a far emergere nelle prime posizioni i documenti rilevanti e contenenti opinioni, a discapito dei documenti solo rilevanti o contenenti solo opinioni.

A livello realizzativo, l'implementazione della metodologia appena richiamata rende necessario l'utilizzo di Terrier [8], l'unico framework per l'IR che, ad oggi, supporta nativamente i modelli di recupero DFR. Terrier è quindi indispensabile sia per la creazione del dizionario che per l'implementazione dell'algoritmo di re-rank. D'altro canto la comunità dell'open-source di Apache Software Foundation[4] sta, già da qualche anno, concentrando le energie sullo sviluppo del crawler Nutch [3] e del framework per motori di ricerca Solr [12]. Tale impegno si concretizza nel frequente rilascio di versioni sempre più stabili e di funzionalità sempre più avanzate. Inoltre, se l'uso di componenti open-source va incontro al requisito non funzionale di economicità dichiarato nella Sezione 1, si evidenzia come la scelta di Nutch permetta di soddisfare anche il requisito di scalabità grazie al suo supporto nativo verso la piattaforma Hadoop [15].

La Figura 1 riporta uno schema architetturale a partire dal quale è possibile descrivere sia le modalità di creazione del dizionario (linee piene etichettate con numeri), sia quelle relative al suo utilizzo (linee tratteggiate contrassegnate da lettere).

Per quanto riguarda la creazione del dizionario, Nutch viene utilizzato per eseguire la strategia di crawling (1) descritta nella Sezione 2. La collezione così prodotta viene indicizzata da Solr (2). Successivamente l'indice viene ripulito (3) per mezzo delle funzionalità di rimozione dei duplicati offerte dalle librerie Lucene [5] . A partire dal contenuto dell'indice viene generato il benchmark (4), secondo le modalità descritte nella Sezione 3, ed esportata una collezione in formato TREC (5) grazie alla quale risulta

[4] http://www.apache.org/

semplice generare un indice Terrier i cui documenti condividono un identificativo comune con i documenti presenti nell'indice della piattaforma Solr. A partire dall'indice Terrier viene infine generato il dizionario (6).

L'algoritmo di re-rank entra in gioco durante la fase di recupero. Più precisamente a fronte di una query eseguita dall'utente (a), il sistema Solr esegue un primo recupero sul proprio indice e inoltra il risultato a Terrier (b). Quest'ultimo esegue l'algoritmo di re-rank (c) e restituisce i risultati a Solr (d) che si incarica di farli visualizzare all'utente (e).

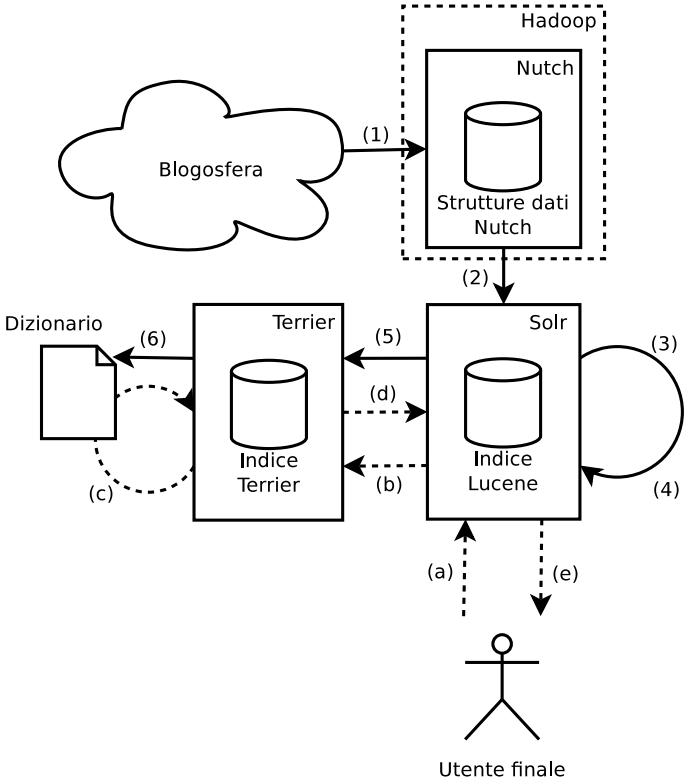


Figura 1. Schema architetturale del prototipo realizzato. Le frecce piene, etichettate con numeri, delineano il processo di creazione del dizionario. Le frecce tratteggiate, etichettare con lettere, mostrano il processo di interrogazione del sistema.

5 Conclusioni e sviluppi futuri

In questo lavoro si riporta l'esperienza maturata nell'applicazione di tecniche di opinion finding al dominio dei blog dedicati ai programmi televisivi trasmessi dalle emittenti italiane. Le attività hanno condotto alla realizzazione di un prototipo, basato su

componenti open-source, in grado non solo di fornire una risposta al problema in questione, ma anche di supportare la creazione di benchmark per l'Information retrieval e la creazione automatica di dizionari italiani composti da termini "opinion-bearing". In tal senso l'intera piattaforma può essere riutilizzata in altri domini applicativi, favorendo sia la realizzazione di nuovi benchmark che la creazione di nuovi dizionari specializzati per i singoli domini.

Riferimenti bibliografici

1. Giambattista Amati, Edgardo Ambrosi, Marco Bianchi, Carlo Gaibisso, and Giorgio Gambosi. Automatic construction of an opinion-term vocabulary for ad hoc retrieval. In Craig Macdonald, Iadh Ounis, Vassilis Plachouras, Ian Ruthven, and Ryen W. White, editors, *ECIR*, volume 4956 of *Lecture Notes in Computer Science*, pages 89–100. Springer, 2008.
2. Gianni Amati and Cornelis Joost Van Rijsbergen. Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM Trans. Inf. Syst.*, 20:357–389, October 2002.
3. Mike Cafarella and Doug Cutting. Building nutch: Open source search. *Queue*, 2:54–61, April 2004.
4. Kushal Dave, Steve Lawrence, and David M. Pennock. Mining the peanut gallery: opinion extraction and semantic classification of product reviews. In *Proceedings of the 12th international conference on World Wide Web*, WWW '03, pages 519–528, New York, NY, USA, 2003. ACM.
5. Lucene. The Lucene search engine, 2005.
6. Craig Macdonald and Iadh Ounis. The TREC Blog06 collection: Creating and analysing a blog test collection. Technical report, Department of Computing Science, University of Glasgow, Scotland, United Kingdom, 2006.
7. Craig Macdonald, Rodrygo L. T. Santos, Iadh Ounis, and Ian Soboroff. Blog track research at TREC. *SIGIR Forum*, 44(1):57–74, 2010.
8. I. Ounis, G. Amati, V. Plachouras, B. He, C. Macdonald, and C. Lioma. Terrier: A High Performance and Scalable Information Retrieval Platform. In *Proceedings of ACM SIGIR'06 Workshop on Open Source Information Retrieval (OSIR 2006)*, 2006.
9. Iadh Ounis, Craig Macdonald, Maarten de Rijke, Gilad Mishne, and Ian Soboroff. Overview of the trec 2006 blog track. In Voorhees and Buckland [14].
10. Bo Pang and Lillian Lee. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1-2):1–135, January 2008.
11. Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up? Sentiment classification using machine learning techniques. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 79–86, 2002.
12. David Smiley and Eric Pugh. *Solr 1.4 Enterprise Search Server*. Packt Publishing, 2009.
13. Ellen M. Voorhees. Overview of the trec 2006. In Voorhees and Buckland [14].
14. Ellen M. Voorhees and Lori P. Buckland, editors. *Proceedings of the Fifteenth Text REtrieval Conference, TREC 2006, Gaithersburg, Maryland, November 14-17, 2006*, volume Special Publication 500-272. National Institute of Standards and Technology (NIST), 2006.
15. Tom White. *Hadoop: The Definitive Guide*. O'Reilly Media, original edition, June 2009.

Tag clouds and retrieved results: The CloudCredo mobile clustering engine and its evaluation

Stefano Mizzaro, Luca Sartori, and Giacomo Strangolino

Department of Mathematics and Computer Science
University of Udine
Udine, Italy
mizzaro@uniud.it, sartori.uni@gmail.com, delleceste@gmail.com

Abstract. We discuss the use of tag clouds as a way of visualizing the results of a clustering search engine. We briefly present a specific tag cloud approach and its implementation in the CloudCredo prototype. Then we describe an experimental user study aimed at demonstrating that tag cloud visualization is: (i) as effective as classical tree like visualization; and (ii) particularly effective on small screen devices. Towards the aim (i), we compare CloudCredo with a similar system, Credino; towards (ii), in the experiment the two systems are compared on iPhone and iPad, two similar devices differing mainly in their size. Results, although preliminar, support the hypotheses.

Keywords: Clustering, Mobile devices, Tagcloud, Evaluation

1 Introduction

On the Web, there is a growing number of *clustering search engines*, namely search (or, more often, meta-search) engines that present the retrieved documents organized in clusters: similar documents are grouped together under a meaningful label; clusters are organized hierarchically (i.e., clusters are divided into sub-clusters, and so on) and usually shown in a tree-like manner; and the end user can browse the retrieved results by focusing on specific clusters. Some examples of these systems are: Yippy (formerly known as Clusty and Vivísimo) www.yippy.com or CREDO credo.fub.it.[1] Even classical search engines like Google show some signal of a clustering approach, although they are still much more oriented towards the classical ranked list.

The cluster approach seems particularly adequate and effective for mobile devices, since it allows to use the limited screen space in a more effective way. This approach has been proposed and evaluated for the CREDO system, and its mobile versions Credino and SmartCREDO [2,3]. Indeed, mobile search engines are an important and hot research topic: as it is well kwon, several statistics

[1] At the time of writing CREDO is not available.

show that Internet traffic in general, and queries to search engines in particular, generated by means of mobile devices are quickly increasing. It is foreseen that by 2015 there will be more mobile users than desktop Internet users.

However, the classical tree-based visualization of document clusters is not the only possibility. In this paper we propose a *tag cloud* based visualization that, in our opinion, has the potential to be particularly effective on small-screen mobile devices. Our aim is twofold:

- to understand if the tag cloud visualization is effective;
- to understand if it is particularly effective on mobile device small screens.

The paper is organized as follows: Sect. 2 defines tag clouds and motivates our approach; Sect. 3 presents CloudCredo, a mobile clustering engine implementing the tag cloud approach, and recalls Credino, a companion system used in the evaluation; Sect. 4 describes the user study that we performed to experimentally evaluate the tag cloud effectiveness.

2 Tag clouds

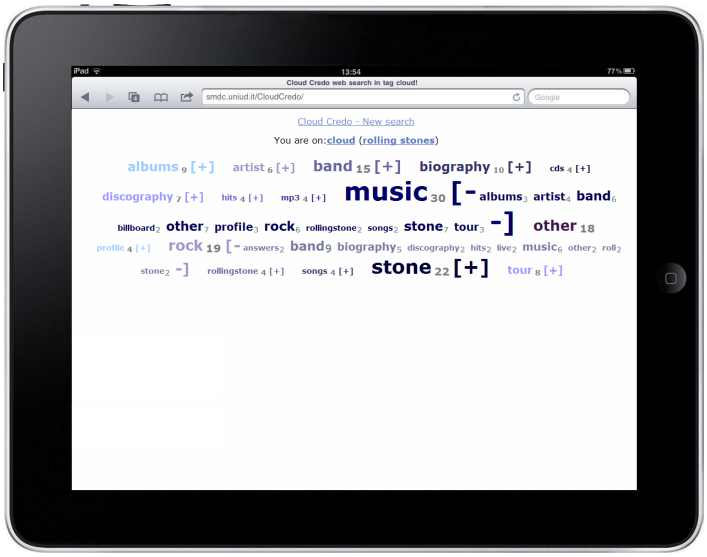
A tag cloud (or word cloud) is a set of terms organized spatially and graphically (in terms of fonts and colors) to visually highlight the most important terms. Tag clouds are very common: they are being used quite often on the Web, to show the tags used to annotate web resources, to summarize the main topics of a Web site, and so on. There are several kinds of tagclouds, that can differ for the selection of terms, the graphical aspect, and the auxiliary information shown (like a count of each term frequency in the original text); a description can be found at `en.wikipedia.org/wiki/Tag_cloud`.

As mentioned above, we propose to use a tag cloud to show the label of the clusters. The rationale for this approach is that a tag cloud can show the same labels and use less space than the classical tree-like visualization, although admittedly in a less organized way. Moreover, not only the cluster labels are shown as a tag cloud, but the labels are clickable, and can be expanded into sub-clusters (as in the tree like visualization). Also, we specifically tailor mobile devices, and we are interested in studying the effectiveness of the tag cloud clustering approach when screen space is limited.

3 Credino and CloudCredo

We build on Credino system [2], implemented with the aim of porting the CREDO clustering engine on a mobile device (a PDA was used in the original paper [2], but we slightly adapted it to more recent devices like the iPhone). Figure 1(a) shows a screenshot of Credino on an iPhone. On the basis of Credino, we implemented CloudCredo, that visualizes the same clusters as Credino by means of a tag cloud. Figures 1(b) and (c) show the screenshot of CloudCredo on an iPhone and an iPad. Credino and CloudCredo are both meta-search engines on CREDO, therefore they both show exactly the same cluster hierarchy,

(a) Credino (b) CloudCredo on iPhone



(c) CloudCredo on iPad

Fig. 1. Credino (a) and CloudCredo (b) on an iPhone with the query [rome]: the cluster "city" has been expanded and the subclusters are shown. CloudCredo (c) on an iPad, with the query [rolling stones], and the clusters "music" and "rock" expanded. Scale factor are different: with the same scale, the iPad would be almost twice bigger.

just visually different. As can be seen in Figures 1(b) and (c), our tag cloud implementation exploits both colors and size, and each cluster also shows the number of documents in it. The tag cloud implementation, similarly to the classical hierarchical tree one, allows to expand a category into subcategories, by clicking on the "`[+]`" sign close to the tag (and to compact it by clicking on "`[-]`"). Both Credino and CloudCredo are Web applications that can be used by any standard Web browser; on iPhone and iPad they adapt smoothly to the portrait/landscape orientation of the device.

We are not alone in proposing to use tag clouds to show the retrieved results; the Quintura search engine `www.quintura.com/` does exactly that. Our approach is slightly different, though, since: (i) our tags/clusters can be expanded into subtags/sub-clusters; and (ii) we specifically target mobile devices in this work.

CloudCredo is available at `smdc.uniud.it/CloudCredo`; the version of Credino used in the experimental evaluation described below is at `credino.dimi.uniud.it/`. The two systems, being based on CREDO (see Footnote 1), are not available at the time of writing.

4 Experimental evaluation

We performed a user study towards the two aims stated at the end of Section 1. These can be translated into the following experimental hypotheses: (i) CloudCredo is as effective as Credino; and (ii) CloudCredo effectiveness turns out to be high in particular on small screens.

4.1 Experimental design

We used the two systems Credino and CloudCredo in our evaluation. We also used an iPhone and an iPad: since the two devices are very similar, the main (if not only) difference being their size, we try in this way to single out the effect of size. Thus, our experiment has two independent variables:

- device, or size (iPhone and iPad);
- system (Credino and CloudCredo).

48 participants, recruited in our university, were involved in our study. Each participant was asked to perform 4 tasks. The tasks were built by starting from the most frequent queries on Google Mobile `www.google.com/intl/en/press/zeitgeist2008/`: we selected 4 of them and built 4 simulated work task situations [1] around them. Figure 2 shows task 1, translated from Italian to English, as given to the user. To have a more controlled environment, we specified the initial query. To limit learning effects, we relied on a Graeco-Latin square design: each subject performed her 4 tasks on the four system/device combinations, in a different order.

As dependent variables, we measured both objective user effectiveness and subjective user satisfaction. User effectiveness was measured as a linear combination of: the success in finding the appropriate page (a binary value in $\{0, 1\}$),

Task 1

- **Description:** Imagine that you are going to visit a friend in Rome and therefore you want to find some information about cultural events (e.g., exhibitions and concerts) that will take place during your stay in town.
- **Task:** Retrieve two different pages.
  A page is relevant if it provides the date, time, and location of an event taking place in Rome during the next 30 days. Pages discussing an event in a general way, without specifying the above data, will be not relevant.
- **Other instructions:** Start with the query [rome events]

Fig. 2. The first task used in the experiment.

the speed (computed on the basis of time needed and normalized into the $[0, 1]$ range), and the confidence the user had to have performed her task correctly (again normalized into $[0, 1]$). We defined three different combinations of these three factors, with different weights; however, there was no difference among the three combinations. In the following we measure effectiveness E as

$$E = \text{success} * (2/3 * \text{speed} + 1/3 * \text{confidence})$$

(if success is 0, then E is 0 as well; speed is more important than confidence).

User satisfaction was measured by means of questionnaires: participants filled in a questionnaire after each task completion, and one final questionnaire as well. Questionnaires collected, by means of Likert scales, data about:

- difficulty of the task;
- difficulty of using the system;
- adequacy of the system to the device.

We combine these three values into a single satisfaction one S' by taking their average, normalized onto $[0, 1]$:

$$S' = 1/3 * \text{task\_d} + 1/3 * \text{system\_d} + 1/3 * \text{adequacy}.$$

We also take into account other two questionnaire items that, as a control, asked whether the participant preferred the other system or device. The final satisfaction S was computed by slightly changing S' to take these into account.

We adopted the usual procedures of a laboratory testing: each subject was briefed and trained, she filled in a first questionnaire with some demographics data, then she started the four task-questionnaire iteration, and finally filled in the last questionnaire. We also ran a pilot test, that confirmed the choice of the four tasks and allowed to estimate the maximum time allowed for each task.

4.2 Results

The collected demographics show that participants were either university students (45 out of 48) or just graduated searching for a job. They had good — and

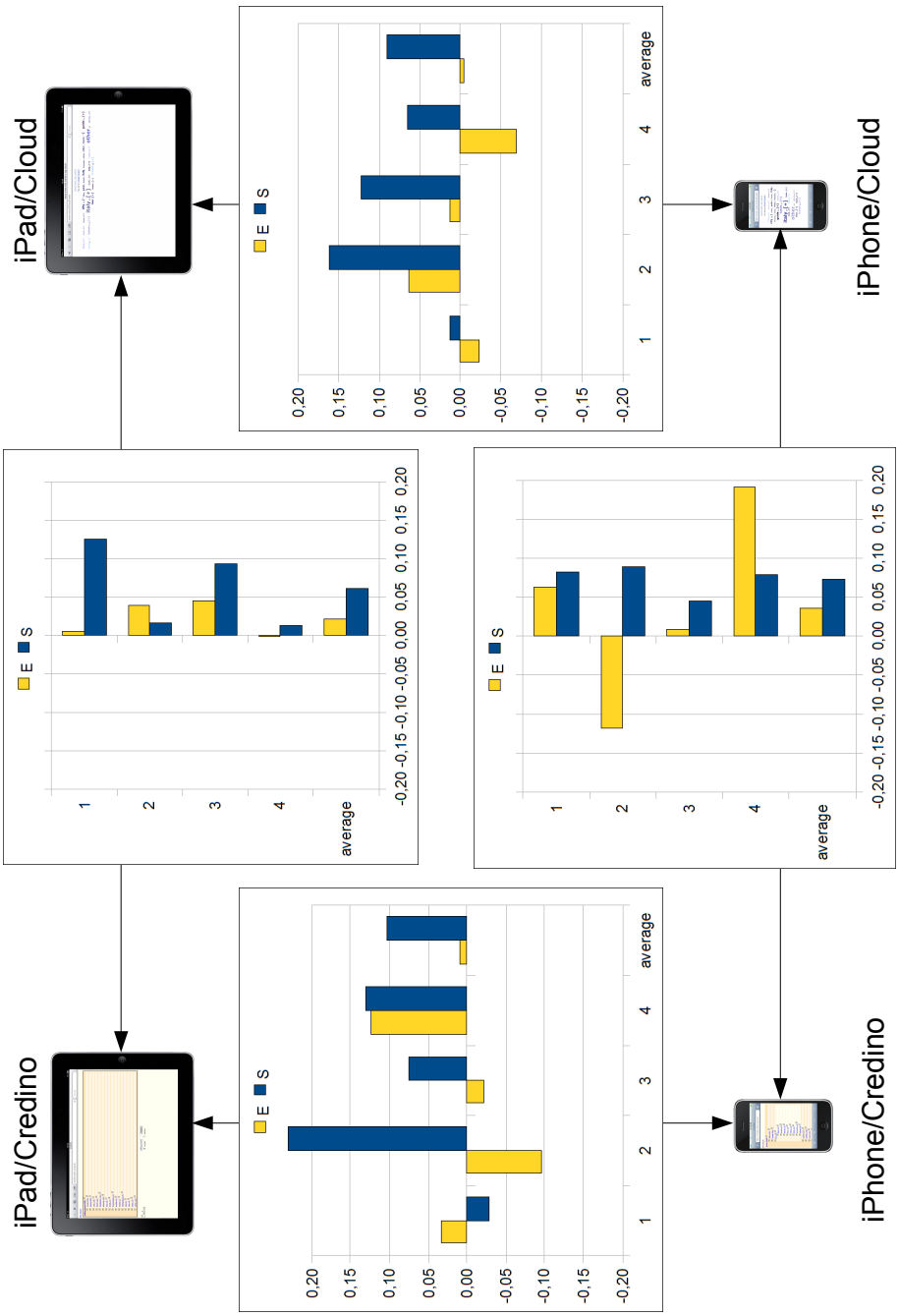


Fig. 3. Overall results.

homogeneous — knowledge of computers, Web search, and mobile devices. All of them were aware of iPhone and iPad devices. Nobody had used a clustering engine before.

Figure 3 shows the overall results. The four device/system combinations are shown in the corners of the figure; the four charts show the differences — in both S and E — for the single tasks and averaged on the four tasks. The bars are oriented towards the best device/system combination, e.g., the leftmost vertical bar shows that iPad/Credino had a higher effectiveness E than iPhone/Credino on task 1.

By analyzing the figure we can understand that:

- Since all the "average" bars point towards right, on average, CloudCredo had both a higher E and a higher S than Credino.
- Since all the average bars point towards up (with a single exception, the rightmost E bar, which is anyway very small in absolute value), on average, the iPad device had both a higher E and a higher S than iPhone.
- Combining the previous two points, iPad/Cloud was the most effective and most preferred combination.
- The above considerations seem stronger for S, which has longer bars.
- We can see that the above results hold for most of the single tasks as well: there are only 6 bars on specific tasks that disagree with the average bar (out of 32 possibilities).
- Also, on the single tasks, E and S are often in agreement, although in 6 out of 16 cases they are not.
- Although we were interested in showing that the tag cloud visualization was as effective as the tree-like one, these results are a first cue that it is even more effective and preferred. However, there is almost no statistical significance on the differences. On E, according to the Mann-Whitney-Wilcoxon test, the only statistically significant difference (at the 0.05 level), is on task 4 between iPhone/Credino and iPhone/Cloud (the longest horizontal E bar in figure). Statistical significance is slightly higher on S: although most of the differences are not significant, the preference of iPad/CloudCredo to iPad/Credino is significant at the 0.05 level, according to the Mann-Whitney-Wilcoxon test.
- Therefore, the two visualization approaches can be considered equivalent, with a slight preference for the tag cloud one. This confirms the first hypothesis.
- Turning to the second hypothesis, the figure shows that the average difference is slightly higher at the iPhone level than at the iPad one. There is no statistical significance for this result, however, also because there are quite high variations over the single topics (i.e., bars on the top chart are often very different from the corresponding bars on the bottom chart — see, for example, the striking difference on the E value on task 2). Thus we can only say that there is a slight indication of the particular effectiveness of the tag cloud approach on small screens, also on the basis of the results in [2,3] that showed how the clustering approach of Credino is more effective on small screens than on large ones.

5 Conclusions

We have proposed a tag cloud based approach to the visualization of the retrieved results by a clustering search engine. Our experimental study on two prototypes supports the hypotheses that tag clouds are an effective visualization alternative, especially on small screen mobile devices. The second point is particularly critical, since we do not have a statistically significant proof of it. We do not have any contrary evidence, though; this, combined with the results of previous studies [2, 3] makes indeed interesting the option of using a tag cloud based approach on mobile devices, although further evidence should be found.

The experimental design needs some further remarks. The usual user study performed in information retrieval aims at demonstrating that a new version of some system reaches higher effectiveness and/or user satisfaction than some baseline. Our experimental study was somehow different from this classical setting, since we were interested in showing that an alternative system (actually, visualization approach) is as effective as a classical one.

Although the results of our user study are positive, they are preliminary: we used four tasks only, and the user population is quite homogeneous. Therefore, a first and obvious future work direction is to repeat the experiments with a higher number of tasks and with a different, and perhaps more heterogeneous, user population. Also, a more sophisticate experimental design can help to prove the second hypothesis. A last direction is to implement native applications for iPhone/iPad (and Android as well) of CloudCredo and Credino: this would allow a more effective interaction and a better user experience.

References

1. P. Borlund. The IIR evaluation model: a framework for evaluation of interactive information retrieval systems. *Information Research*, 8(3):paper no. 152+, 2003.
2. C. Carpineto, A. Della Pietra, S. Mizzaro, and G. Romano. Mobile Clustering Engine. In *Proceedings of the 28th European Conference on Information Retrieval, London, UK*, volume 3936 of *Lecture Notes in Computer Science*, pages 155–166. Springer, 2006.
3. C. Carpineto, S. Mizzaro, G. Romano, and M. Snidero. Mobile information retrieval with search results clustering: Prototypes and evaluations. *Journal of the American Society for Information Science and Technology*, 60(5):877–895, 2009.

The Collaboration Potential, an index to assess the roles of scientists in their coauthorship networks

Francesco Giuliani[1], Michele Pio De Petris[1] and Giovanni Nico[2]

[1] Innovation and Technological Development, IRCCS Casa Sollievo della Sofferenza, San Giovanni Rotondo, Italy,
f.giuliani@operapadrepio.it, m.depetris@operapadrepio.it,
[2] Istituto per le Applicazioni del Calcolo, Consiglio Nazionale delle Ricerche (CNR-IAC), Italy, g.nico@ba.iac.cnr.it

Abstract. Over the past decade there have been many investigations aimed at defining the role of scientists in their coauthorship networks. In this work we propose an analytical definition of a collaboration potential between authors of scientific papers based on both coauthorships and content sharing. The collaboration potential can also be considered a tool to investigate the weakness of the network in terms of 'lost collaborations' between authors with same scientific interests. This work is an abbreviated version of the original article from the same authors [1].

Keywords: social network analysis, scientific collaborations, coauthorships, collaboration potential

1 Introduction and methodological approach

In this work we present a method aimed at investigating the informative potential that modern bibliographic databases offer. We study the publication output of researchers and try to find an index describing both collaborations and content sharing in scientific networks.

Considering coauthorship as synonymous of collaboration, we can define a collaboration index between author A and author B as

$$P_{AB} = \frac{dim(P_A \cap P_B)}{dim(P_A)} \ , \tag{1}$$

where P_A and P_B represent the sets of papers authored by A and B respectively, $dim(P_A)$ represents the number of elements (articles) authored by author A and $dim(P_A \cap P_B)$ represents the number of articles shared by authors A and B as coauthors. This index represents for author A the fraction of articles he has written in collaboration with author B. This index, taken alone, does not tell the whole story about collaboration as it is independent from article contents.

One can build an index to express content sharing defining it as the number of keywords author A and author B share divided by the number of keywords of

author A. This index is a measure of the commonality of scientific interests, but does not take into account collaborations between authors. If we want to measure the collaboration potential between two authors we need to build a consistent index taking into account both coauthored papers and contents of such papers, that in our model are represented by article keywords.

We want keywords to come from an unambiguous and limited set of terms, so we chose to study only publications indexed by the PubMed search engine. For such publications, keywords come from MeSH (Medical Subject Headings) database, a controlled vocabulary thesaurus used for indexing articles in PubMed. We simply used a custom query and XML parsing in order to associate keywords to articles of our interest.

2 Measuring the collaboration potential

In our simple model we start taking coauthorships into account. We can define the set of articles author A has not co-authored with author B (and vice versa) as

$$\overline{P}_A = P_A - (P_A \cap P_B) \ , \ \overline{P}_B = P_B - (P_A \cap P_B) \tag{2}$$

The articles belonging to these sets are associated with their respective keywords, i.e. we can define the sets $\overline{K}_A$ and $\overline{K}_B$ containing the keywords of the papers the two authors have not respectively coauthored. The intersection between these two sets

$$\overline{K}_{AB} = \overline{K}_A \cap \overline{K}_B \ , \tag{3}$$

represents the keywords shared by the articles the authors have not co-authored. So we can formulate the collaboration potential based on non-coauthorship for author A towards author B as

$$m_{AB} = \frac{\dim(\overline{K}_A \cap \overline{K}_B)}{\dim(\overline{K}_A)} \ . \tag{4}$$

This index has many interesting characteristics. It is defined in the interval $[0, 1]$ and is 0 in two circumstances:

- First case: the two authors have coauthored all their articles. In this case the sets $\overline{K}_A$ and $\overline{K}_B$ are both void so $\dim(\overline{K}_A \cap \overline{K}_B)$ is 0, the authors having fully exploited their collaboration potential, having co-authored all they could, i. e. all the articles they wrote.
- Second case: for the articles they have not coauthored, they worked on totally different subjects. In this case $\overline{K}_A \cap \overline{K}_B$ is void meaning that the authors, excluding coauthored articles, share no common scientific interests and so, according to our model, no collaboration potential exists between them.

We can discriminate between the two cases in which $m_{AB} = 0$ according to the corresponding value of P_{AB}. In fact a value $P_{AB} = 1$ corresponds to the first case, while a $P_{AB} \neq 1$ to the second case.

In all other cases m_{AB} different from 0 implies the existence of a not fully "exploited" collaboration between authors A and B. The other extreme value of the index is 1. In this case $\overline{K}_A = \overline{K}_{AB} = \overline{K}_B$, i.e. author A and author B share all their keywords for the articles they have not coauthored. It is worth noting that the collaboration potential we've just defined should not be considered a "predictor" of future collaborations but it is intended to investigate the role of scientists in the collaboration network. Other approaches were proposed in the literature [2], [3] and methods were presented in order to predict the evolution of links in a social network based on topology taken alone. Our method is quite different because it relies on intrinsic node properties (identified as keywords), and tries to investigate properties of links in terms of 'lost collaboration' between the authors.

Extending 4 we can easily compute the collaboration potential between author A and group G considering the group as a single author, i. e. considering the set of articles written by author A and the set of articles written by all other authors of group G. We thus obtain:

$$m_{AG} = \frac{\dim(\overline{K}_A \cap \overline{K}_G)}{\dim(\overline{K}_A)} \,, \tag{5}$$

where $\overline{K}_G$ represents the set of keywords for the articles author A has not coauthored with the other authors belonging to group G. If author A belongs to group G the index in (5) expresses the collaboration potential the author has with the colleagues of his own group, supposedly studying the same subjects of his researches and publications.

3 Application of the method and discussion of the results

To apply our method we considered the publications and authors of the Casa Sollievo della Sofferenza research hospital in years 2004-2009. For all authors (216) and publications (711 papers, with a mean of 14.42 keywords per article) we computed the collaboration potential according to eqs (4) and (5). We found a mean value for P_{AG} of 90.50%, confirming that scientists coauthor the largest majority of their publications with authors of their own group than with authors belonging to other research groups of the institute.

In order to investigate the role of scientists inside and between research groups, we considered the values of P_{AG} and of m_{AG} for each researcher (see fig.1). The majority of authors concentrate on the bottom-right area of the plot. This result confirms that generally authors have a low collaboration potential with colleagues of their groups. The value of the collaboration potential is exactly zero for 81.48% of authors. This result is simply understandable in terms of coauthorships, in fact we have found that in all cases in which m_{AG} is zero P_{AG} is one, meaning that each of these authors' publication is coauthored by at least one other author of the group the author belongs to. We could define these authors as highly integrated with their research units, writing their papers with

at least one of the colleagues of their groups. Furthermore, we found a small subset of authors having a low value of P_{AG} and an high value of m_{AG} with their group. We can easily define these authors as "independent" as they share no article with the members of their own group, given many subjects on which they "could" have written articles together.

Eventually, generalizing the concept of collaboration to a broader scope, the methods presented herein could easily be used to define a collaboration potential in every case in which one can classify the content of some activities and determine which of them are in common among the actors cooperating to perform such activities.

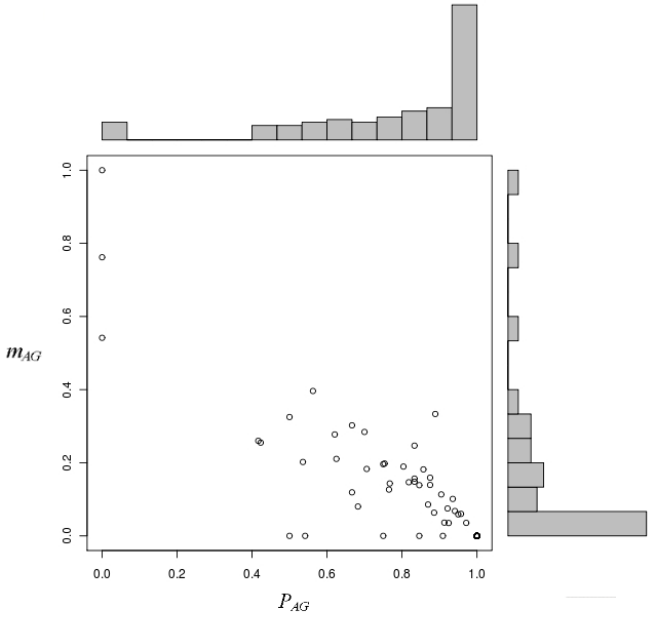


Fig. 1. Distribution of authors according to the collaboration potentials toward their research groups (m_{AG}) and coauthorship sharing (P_{AG}) values. The grey bar graphs on the axes show the frequency distributions of the number of authors for each interval of (m_{AG}) and (P_{AG}).

References

1. *Giuliani F, De Petris MP, Nico G*, Assessing scientific collaboration through coauthorship and content sharing, *Scientometrics*, 85(1), p13–28, October 2010.
2. *David Liben-Nowell and Jon Kleinberg*, The Link-Prediction Problem for Social Networks, *Journal of the American Society for Information Science and Technology*, 58(7), p1019–1031, May 2007.
3. *Leo Katz*, A new status index derived from sociometric analysis, *Psychometrika*, 18(1), p39–43, March 1953.

Strategie di classificazione per servizi di search della Pubblica Amministrazione

Marco Bianchi[1], Mauro Draoli[2] Giorgio Gambosi[1], Alessandro Ligi[2], and Marco Serrago

[1] University of Rome "Tor Vergata",
Via della Ricerca Scientifica 1, 00133 Rome, Italy
bianchi@mat.uniroma2.it
gambosi@mat.uniroma2.it
[2] DigitPA, Viale Marx 43 - 00137 Rome, Italy
draoli@digitpa.gov.it
alessandro.ligi@digitpa.gov.it

Sommario In questo lavoro si introducono le attività sperimentali finalizzate alla realizzazione di un servizio per la ricerca della modulistica pubblicata dalle Pubbliche Amministrazioni (PA) italiane sui propri siti istituzionali e condotte nell'ambito del progetto pubblico "Italia.gov.it - il motore della PA digitale". In tale contesto la necessità di creare e aggiornare una collezione composta da soli moduli rende necessaria l'introduzione di classificatori automatici che siano in grado di supportare il filtering della grande mole di documenti che vengono recuperati a valle dell'attività di crawling. Il caso presentato è interessante perchè mostra quanto la scelta del classificatore da adottare possa essere influenzata dai vincoli economici e organizzativi tipicamente posti dalle Pubbliche Amministrazioni.

Keywords: vertical search engine, classification, active learning.

Oggigiorno la realizzazione di servizi di *search* verticali per il dominio della Pubblica Amministrazione (PA) è un'attività il cui valore scientifico, economico e sociale viene riconosciuto a livello internazionale. Il motore di ricerca USA.GOV[3] è forse il principale esempio di applicazioni in esercizio con l'obiettivo di supportare i cittadini e le aziende nella ricerca di informazioni e documenti pubblicati sul Web dalla PA. Anche in Italia è stato avviato un progetto pubblico finalizzato alla realizzazione di un motore di ricerca della PA, denominato Italia.gov.it[4]. Nell'ambito di questo progetto ogni funzionalità di *search* definisce, di fatto, un *task* a sé stante che spesso richiede la sperimentazione di soluzioni innovative.

In questo lavoro si illustrano alcune problematiche che si stanno affrontando durante le attività sperimentali finalizzate alla realizzazione di un servizio per la ricerca della modulistica pubblicata dalle PA sui propri siti istituzionali. Tale servizio, denominato *moduli-on-line*, rappresenta un esempio significativo

[3] http://search.usa.gov/
[4] http://www.italia.gov.it

delle funzionalità di *search* erogate da Italia.gov.it e di come esse possono essere implementate.

Uno degli aspetti innovativi di Italia.gov.it risiede nella presenza di una base di conoscenza da cui si attingono tutte le informazioni che vengono indicizzate per la realizzazione dei singoli servizi di *search*. Tale base di conoscenza è caratterizzata da una modalità di aggiornamento automatico eseguita per mezzo di strumenti di Information Retrieval e Text Mining allo stato dell'arte. Nello specifico, il servizio moduli-on-line indicizza tutti i documenti scoperti sul Web per mezzo di una continua attività di crawling e marcati come *moduli* da classificatori binari precedentemente addestrati. Il lavoro svolto dai classificatori è pertanto determinante per ottenere una buona qualità degli indici di ricerca: se i classificatori svolgono il proprio lavoro con precisione i risultati presentati agli utenti saranno composti, per lo più, da modulistica, viceversa saranno "inquinati" da errori di classificazione (falsi positivi) che potrebbero minare alla base la fiducia sul funzionamento del sistema. Poiché la qualità del servizio che si intende realizzare è così fortemente influenzata dal funzionamento dei classificatori, è stata avviata un'attività finalizzata alla creazione di un benchmark utile per l'addestramento, la misurazione e la scelta del miglior tipo di classificatore per il problema in oggetto.

In questo lavoro si descrive la strategia che si sta studiando per la gestione del servizio moduli-on-line, conciliando esigenze di natura sia economica che tecnica. Per quanto riguarda le esigenze economiche, l'architettura del sistema Italia.gov.it, descritta in [1], prevede il possibile intervento di esperti di dominio (oracoli, dal punto di vista del processo di classificazione) che da un lato eseguono un monitoraggio continuo sulla precisione del sistema di classificazione, dall'altro risolvono i casi di incertezza degli strumenti di classificazione contribuendo, di fatto, a un arricchimento del training set [5]. In questo scenario, si richiede che, in fase di produzione, gli oracoli siano messi nella condizione di classificare quanti più moduli possibile, al fine di massimizzare il numero di documenti indicizzati. È evidente, infatti, come la classificazione di un non-modulo non sia immediatamente riutilizzabile in fase di produzione. Dal punto di vista tecnico-scientifico, invece, il training set deve essere rappresentativo del dominio di classificazione e il suo aggiornamento deve essere finalizzato al miglioramento delle prestazioni dei classificatori. Pertanto, anche gli esempi veri negativi e falsi positivi possono essere utili per migliorare il sistema di classificazione. È obiettivo della sperimentazione in corso verificare la compatibilità tra le due esigenze.

L'attività di sperimentazione è condotta concentrando l'attenzione su classificatori di tipo Support Vector Machine (SVM) [2], Naive Bayes (NB) [4], Logistic Regression (LR) [3] e Dynamic Language Model (DLM) [6].

La prima versione del benchmark di moduli-on-line, è composta da 8475 documenti recuperati a valle di un'attività di crawling eseguita nel mese di marzo 2011 su 12 siti istituzionali di PA centrali indicati da esperti di dominio. Il crawler è stato configurato in modo da scaricare solamente documenti con estensioni .pdf, .doc, .docx, .rtf, .xls e .xlsx in quanto si pensa possano essere i principali formati utilizzati dalle PA per la pubblicazione della modulistica. L'intero insieme dei

file è stato successivamente classificato a mano da esperti di dominio che hanno individuato 793 documenti appartenenti alla categoria dei moduli e 7461 a quella dei non-moduli. Per la fase di valutazione si è considerato come *modulo* ogni documento testuale realizzato per scopi amministrativi e burocratici comprensivo di una serie di campi compilabili da un generico utente. Nella classe complementare sono invece stato inseriti tutti gli altri documenti. Esempi tipici di documenti classificati come non-moduli sono le Determinazioni Dirigenziali, le Disposizioni Direttoriali, le Leggi, i Decreti Legge, gli Avvisi Pubblici. Si evidenzia la presenza di casi che potrebbero essere considerati di ambiguità. Esistono infatti documenti che sono caratterizzati dalla presenza di una prima parte documentale, e una seconda parte compilabile. Un esempio di questa tipologia di documenti è un Bando di Concorso che è tipicamente composto da un certo numero di articoli che regolamentano la procedura concorsuale e da alcuni modelli di modulo in appendice. Su indicazione degli esperti di dominio, i casi di incertezza sono stati aggiunti alla categoria dei moduli.

Un primo training set delle dimensioni di 547 documenti (composto da 325 documenti e 222 moduli) è stato costruito da esperti di dominio. La Tabella 1 riporta le prestazioni dei classificatori presi in esame sul test-set composto dai rimanenti 7707 documenti.

Classif.	Precision	FP-rate	Recall	Accuracy	F-measure
DLM	0.88	0.01	0.82	0.97	0.84
LR	0.79	0.02	0.58	0.92	0.67
SVM	0.66	0.04	0.70	0.94	0.68
NB	0.51	0.05	0.71	0.93	0.60

Tabella 1. Tabella che riassume le prestazioni dei classificatori analizzati. La stabilità dei risultati è stata verificata mediante cross-validation.

A una prima analisi, il classificatore che sembra fornire le migliori prestazioni è il DLM con una precisione dell'88% e una recall dell'82%. Purtroppo però, per requisiti di progetto, la precisione di classificazione deve superare il 95% anche a costo di coinvolgere gli oracoli nel processo di classificazione. Di conseguenza il problema diventa individuare quell'insieme di documenti da sottomettere agli oracoli al fine di superare il 95% di precisione, non penalizzando eccessivamente la recall e minimizzare il lavoro manuale svolto dagli oracoli.

Per affrontare questo nuovo problema si è pensato di osservare i valori di probabilità che i singoli classificatori assegnano ai documenti al fine di fornire indicazione sul grado di confidenza con il quale determinano l'appartenenza a una classe. La Tabella 2 mostra alcuni dettagli sul comportamento dei classificatori DLM e SVM. Analizzando gli insiemi dei documenti classificati come moduli si osserva come il classificatore DLM, assegni la classe di appartenenza con probabilità maggiore del 95% in ben 7680 casi; diversamente SVM ha un comportamento che potremmo definire più cauto, assegnando solo in 3786 casi una probabilità maggiore del 95%. Fissato un intervallo $(x-5, x]$ consideriamo il seguente processo semi-automatico di classificazione:

1. Tutti i documenti identificati come moduli dal classificatore con probabilità maggiore di $x\%$ sono accettati come tali;

2. I documenti classificati come moduli con probabilità compresa tra 50% e x% sono sottoposti alla valutazione manuale assumendo che tale valutazione abbia errore nullo.

Denotiamo come $P_{inc}(x)$ la precisione derivante da questo processo semi-automatico di classificazione. Le stesse considerazioni possono essere effettuate riguardo la recall indicata con $R_{inc}(x)$. La Tabella 2 mostra come considerando una strategia di classificazione semi-automatica, il classificatore SVN sia da considerarsi preferibile in quanto mette il decisore finale nella condizione di poter "acquistare" la precisione voluta pagando il costo di valutazione manuale delle classificazioni incerte (es. 466 valutazioni per raggiungere il 95.7% di precisione). Si evidenzia che se si è disposti a perdere in recall, la precisione può addirittura aumentare riducendo il costo di valutazione manuale. Ciò è possibile passando agli oracoli un pacchetto di documenti selezionato negli intervalli di confidenza con probabilità più alta. È obiettivo di questa sperimentazione verificare che questa modalità di scelta dei documenti da passare agli oracoli non penalizzi le prestazioni dei classificatori successivamente alle attività di ri-addestramento. A tal fine si è deciso di incrementare sensibilmente la dimensione del benchmark, che sarà ampliato fino a raggiungere circa 30.000 documenti classificati a mano.

	(50,55]	(55,60]	(60,65]	(65,70]	(70,75]	(75,80]	(80,85]	(85,90]	(90,95]	(95,100]
DLM										
TP+FP	1	2	0	0	2	0	1	2	3	841
TN+FN	0	3	2	2	0	1	2	3	5	6839
P_{inc}	0.876	0.877	0.877	0.877	0.878	0.878	0.878	0.878	0.879	1
R_{inc}	0.816	0.819	0.819	0.819	0.821	0.821	0.822	0.825	0.827	1
SVM										
TP+FP	98	68	60	47	46	50	31	36	30	284
TN+FN	99	150	199	237	316	415	584	620	837	3502
P_{inc}	0.706	0.749	0.785	0.815	0.846	0.880	0.914	0.940	0.957	1
R_{inc}	0.782	0.846	0.881	0.902	0.926	0.954	0.958	0.969	0.972	1

Tabella 2. Dettaglio dei comportamenti dei classificatori DLM e SVM.

Riferimenti bibliografici

1. Bianchi, M., Draoli, M., and Gambosi, G. An innovative approach to the development of e-government search services. In *EGOVIS* (2011), K. N. Andersen, E. Francesconi, Å. Grönlund, and T. M. van Engers, Eds., vol. 6866 of *Lecture Notes in Computer Science*, Springer, pp. 41–55.
2. Cortes, C., and Vapnik, V. Support-vector networks. *Machine Learning 20* (1995), 273–297. 10.1007/BF00994018.
3. Hosmer, D. W., and Lemeshow, S. *Applied logistic regression (Wiley Series in probability and statistics)*, 2 ed. Wiley-Interscience Publication, 2000.
4. John, G., and Langley, P. Estimating continuous distributions in bayesian classifiers. In *In Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence* (1995), Morgan Kaufmann, pp. 338–345.
5. Settles, B. Active learning literature survey. Tech. rep., 2010.
6. Zhai, C. *Statistical Language Models for Information Retrieval*. Now Publishers Inc., Hanover, MA, USA, 2008.