Knowledge Extraction and Consolidation from Social Media (KECSM 2012)

Preface

In this new information age, where information, thoughts and opinions are shared so prolifically through online social networks, tools that can make sense of the content of these networks are paramount. In order to make best use of this information, we need to be able to distinguish what is important and interesting, and how this relates to what is already known. Social web analysis is all about the users who are actively engaged and generate content. This content is dynamic, rapidly changing to reflect the societal and sentimental fluctuations of the authors as well as the ever-changing use of language. While tools are available for information extraction from more formal text such as news reports, social media affords particular challenges to knowledge acquisition, such as multilinguality not only across but within documents, varying quality of the text itself (e.g. poor grammar, spelling, capitalisation, use of colloquialisms etc), and greater heterogeneity of data. The analysis of non-textual multimedia information such as images and video offers its own set of challenges, not least because of its sheer volume and diversity. The structuring of this information requires the normalization of this variability by e.g. the adoption of canonical forms for the representation of entities, and a certain amount of linguistic categorization of their alternative forms.

Due to the reasons described above, data and knowledge extracted from social media often suffers from varying, non-optimal quality, noise, inaccuracies, redundancies as well as inconsistencies. In addition, it usually lacks sufficient descriptiveness, usually consisting of labelled and, at most, classified entities, which leads to ambiguities.

This calls for a range of specific strategies and techniques to consolidate, enrich, disambiguate and interlink extracted data. This in particular benefits from taking advantage of existing knowledge, such as Linked Open Data, to compensate for and remedy degraded information. A range of techniques are exploited in this area, for instance, the use of linguistic and similarity-based clustering techniques or the exploitation of reference datasets. Both domain-specific and cross-domain datasets such as DBpedia or Freebase can be used to enrich, interlink and disambiguate data. However, case- and content-specific evaluations of quality and performance of such approaches are missing, hindering the wider deployment. This is of particular concern, since data consolidation techniques involve a range of partially disparate scientific topics (e.g. graph analysis, data mining and interlinking, clustering, machine learning), but need to be applied as part of coherent workflows to deliver satisfactory results.

The KECSM 2012 workshop aims to gather innovative approaches for knowledge extraction and consolidation from unstructured social media, in particular from degraded user-generated content (text, images, video) such as tweets, blog posts, forums and user-generated visual media. KECSM has gathered novel works from the fields of data analysis and knowledge extraction, and data enrichment, interlinking and consolidation. Equally, consideration has been given to the application perspective, such as the innovative use of extracted knowledge to navigate, explore or visualise previously unstructured and disparate Web content.

KECSM 2012 had a number of high-quality submissions. From these, the 8 best papers were chosen for the two paper sessions of the programme. To initiate the workshop, a keynote on perspectives of social media mining from an industry viewpoint was given by Seth Grimes.

We sincerely thank the many people who helped make KECSM 2012 such a success: the Program Committee, the paper contributors, and all the participants present at the workshop. In addition, we would like to add a special note of appreciation for our keynote speaker, Seth Grimes, and the ARCOMEM project (http://www.arcomem.eu) for funding the best paper prize.

Diana Maynard
Stefan Dietze
Wim Peters
Jonathon Hare

Organisation

Organising Committee

Diana Maynard, University of Sheffield, United Kingdom
Stefan Dietze, L3S Research Centre, Leibniz University Hannover, Germany
Wim Peters, University of Sheffield, United Kingdom
Jonathon Hare, University of Southampton, United Kingdom

Program Committee

Harith Alani, The Open University, United Kingdom
Sören Auer, University of Leipzig, Germany
Uldis Bojar, University of Latvia, Latvia
John Breslin, NUIG, Ireland
Mathieu D'Aquin, The Open University, United Kingdom
Anita de Waard, Elsevier, The Netherlands
Adam Funk, University of Sheffield, United Kingdom
Daniela Giordano, University of Catania, Italy
Alejandro Jaimes, Yahoo! Research Barcelona, Spain
Paul Lewis, University of Southampton, United Kingdom
Véronique Malaisé, Elsevier, The Netherlands
Pavel Mihaylov, Ontotext, Bulgaria
Wolfgang Nejdl, L3S Research Centre, Leibniz University Hannover, Germany
Thomas Risse, L3S Research Centre, Leibniz University Hannover, Germany
Matthew Rowe, The Open University, United Kingdom
Milan Stankovic, Hypios & Universit Paris-Sorbonne, France
Thomas Steiner, Google Germany, Germany
Nina Tahmasebi, L3S Research Centre, Leibniz University Hannover, Germany
Raphael Troncy, Eurecom, France
Claudia Wagner, Joanneum Research, Austria

Keynote Speaker

Seth Grimes, Alta Plana Corporation, USA

Sponsors

The best paper award was kindly sponsored by the European project ARCOMEM (http://arcomem.eu).

Table of Contents

A Platform for Supporting Data Analytics on Twitter: Challenges and Objectives[1]

Yannis Stavrakas Vassilis Plachouras

IMIS / RC "ATHENA"
Athens, Greece

{yannis, vplachouras}@imis.athena-innovation.gr

Abstract. An increasing number of innovative applications use data from online social networks. In many cases data analysis tasks, like opinion mining processes, are applied on platforms such as Twitter, in order to discover what people think about various issues. In our view, selecting the proper data set is paramount for the analysis tasks to produce credible results. This direction, however, has not yet received a lot of attention. In this paper we propose and discuss in detail a platform for supporting processes such as opinion mining on Twitter data, with emphasis on the selection of the proper data set. The key point of our approach is the representation of term associations, user associations, and related attributes in a single model that also takes into account their evolution through time. This model enables flexible queries that combine complex conditions on time, terms, users, and their associations.

Keywords: Social networks, temporal evolution, query operators.

1 Introduction

The rapid growth of online social networks (OSNs), such as Facebook or Twitter, with millions of users interacting and generating content continuously, has led to an increasing number of innovative applications, which rely on processing data from OSNs. One example is opinion mining from OSN data in order to identify the opinion of a group of users about a topic. The selection of a sample of data to process for a specific application and topic is a crucial issue in order to obtain meaningful results. For example, the use of a very small sample of data may introduce biases in the output and lead to incorrect inferences or misleading conclusions. The acquisition of data from OSNs is typically performed through APIs, which support searching for keywords or specific user accounts and relationships between users. As a result, it is not straightforward to select data without having an extensive knowledge of related keywords, influential users and user communities of interest, the discovery of which is a manual and time-consuming process. Selecting the proper set of OSN data is important not only for opinion mining, but for data analytics in general.

[1] This work is partly funded by the European Commission under ARCOMEM (ICT 270239).

In this paper, we propose a platform that makes it possible to manage data analysis campaigns and select relevant data from OSNs, such as Twitter, based not only on simple keyword search, but also on relationships between keywords and users, as well as their temporal evolution. Although the platform can be used for any OSN having relationships among users and user posts, we focus our description here on Twitter. The pivotal point of the platform is the model and query language that allow the expression of complex conditions to be satisfied by the collected data. The platform models both the user network and the generated messages in OSNs and it is designed to support the processing of large volumes of data using a declarative description of the steps to be performed. To motivate our approach, in what follows we use opinion mining as a concrete case of data analysis, however the proposed platform can equally support other analysis tasks. The platform has been inspired by work in the research project ARCOMEM[2], which employs online social networks to guide archivists in selecting material for preservation.

2 Related Work

There has been a growing body of work using OSN data for various applications. Cheong and Ray [5] provide a review of recent works using Twitter. However, there are only few works that explore the development of models and query languages for describing the processing of OSN data. Smith and Barash [4] have surveyed visualization tools for social network data and stress the need for a language similar to SQL but adapted to social networks. San Martín and Gutierrez [3] describe a data model and query language for social networks based on RDF and SPARQL, but they do not directly support different granularities of time. Mustafa et al. [2] use Datalog to model OSNs and to apply data cleaning and extraction techniques using a declarative language. Doytsher et al. [1] introduced a model and a query language that allow to query with different granularities for frequency, time and locations, connecting the social network of users with a spatial network to identify places visited frequently by users. However, they do not consider any text artifacts generated by users (e.g. comments posted on blogs, reviews, tweets, etc.).

The platform we propose is different from the existing works in that we incorporate in our modeling the messages generated by users of OSNs and temporally evolving relationships between terms, in addition to relationships between users. Moreover, we aim to facilitate the exploration of the context of data and enable users to detect associations between keywords, users, or communities of users.

3 Approach and Objectives

We envisage a platform able to adapt to a wide spectrum of thematically disparate opinion mining campaigns (or data analysis tasks in general), and provide all the in-

[2] http://www.arcomem.eu/

frastructure and services necessary for easily collecting and managing the data. This platform is depicted in Fig. **1**, and comprises three layers.

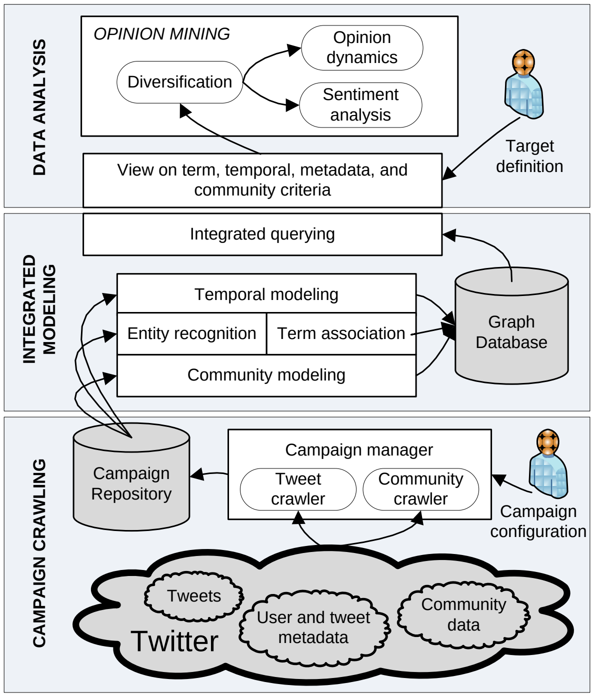


Fig. 1: Platform architecture

The first layer is the *Campaign Crawling* layer. The objective of this layer is to allow for the definition and management of *campaigns*, and to collect all the relevant “raw” data. A campaign is defined by a set of filters and a timespan. For each campaign, the platform monitors the Twitter stream for the duration specified by the campaign’s timespan, and incrementally stores in the *Campaign Repository* all the data that match the campaign filters. Stored data fall into three categories: tweets, metadata, and community data. Community data describe the relationships among Twitter users, while metadata may refer either to tweets (timestamp, location, device, language, etc.) or to users (place, total activity, account creation date, etc.). Selecting the correct tweets for a campaign is of paramount importance; therefore, for converging to the proper filters the process of *campaign configuration* follows an iterative approach: a preliminary analysis on an initial small number of tweets gives an overview of the expected results, indicates the most frequent terms, and highlights the most influential users, in order to verify that the campaign is configured correctly. If this is not the case, an adjustment step takes place by modifying the terms that tweets are expected to contain, the user accounts deemed most relevant to the topic of the campaign, or by removing tweets from user accounts that have been identified to be robots. Those modifications are based on suggestions made by the platform according to the preliminary analysis of tweets.

The second layer is the *Integrated Modeling* layer. The objective of this layer is to support a complex and flexible querying mechanism on the campaign data, allowing the definition of *campaign views*. The significance of campaign views will become apparent in a motivating example that will follow. A prerequisite for querying is the modeling of the campaign "raw" data. The model should encompass three dimensions. First, represent associations between interesting terms contained in tweets. Such terms can be hashtags, or the output of an entity recognition process. Associations between terms can be partly (but not solely) based on co-occurrence of the terms in tweets. Second, represent associations between users of varying influence forming communities that discuss distinct topics. Finally, it should capture the evolution across time of the aforementioned associations, term attributes, and user attributes. This temporal, often overlooked, dimension is of paramount importance in our approach since it enables the expression of time-aware conditions in queries. The resulting model integrates all three dimensions above in a unified way as a graph. A suitable query language is then used to select the tweets that satisfy conditions involving content, user, and temporal constraints.

The third layer is the *Data Analysis* layer. The query language is used to define a "target" view on the campaign data. This "target" view corresponds to a set of tweets that hopefully contain the answer to an opinion-mining question. This set is then fed into a series of analysis processes, like *diversification* for ensuring the correct representation of important attribute values, and *sentiment analysis* for determining the attitude of users towards the question. *Opinion dynamics* is important for recognizing trends across time.

Motivating example. A marketing specialist wants to learn what people say in Twitter about Coca-Cola: (a) in cases when Pepsi-Cola is also mentioned, and (b) during the Olympic Games. The first step is to define a campaign. He launches a preliminary search with the keywords *coca cola*. An initial analysis of the first results reveals other frequent terms and hashtags, and after reviewing them he decides to include the following in the campaign definition: *#cc*, *#cola* and *coke*. Moreover, the platform suggests groups of users whose tweets contain most often the relevant keywords. He decides to include some of them in the campaign definition. Having set the campaign filters, he sets the campaign timespan, and launches the campaign. The crawler downloads data periodically, which are incrementally stored in the Campaign Repository and modeled in the Graph Database. The next important step for our marketing specialist is to define suitable "targets" that correspond to his initial questions. He does that by using the query language of the platform for creating views on the campaign data. An intuitive description of the queries for the two cases follows.

The first query returns the tweets that will hopefully reveal what people say about Coca-Cola in cases when Pepsi is also mentioned, using the following steps: 1) find the terms that are highly associated with Pepsi Cola, 2) return the tweets in which Coca- and Pepsi-related terms are highly associated.

The second query return the tweets that will hopefully reveal what people say about Coca-Cola during the Olympics in the following steps: 1) find the terms that are highly associated with Olympic Games, 2) find the time periods during which those terms are most often encountered, 3) find the groups of people that most often use those terms during the specific time periods, 4) return the tweets of those people, during the specified time periods, which mention the Coca-Cola search terms.

The final step is to conduct opinion-mining analysis on the two sets of tweets returned by the queries. In our approach the emphasis is on selecting the most suitable set of tweets, according to the question we want to answer. In our view, selecting the proper set is paramount for opinion mining processes to produce credible results.

4 Technical and Research Challenges

There are several technical and research challenges that need to be addressed in the implementation of the proposed platform, with respect to the acquisition of data, as well as the modeling and querying of data.

Scalable Crawling. In the case that the platform handles multiple campaigns in parallel, there is a need to optimize the access to the OSN APIs, through which data is made available. Typically, APIs have restrictions in the number of requests performed in a given time span. The implementation of the platform should aim to minimize the number of API requests while fetching data for many campaigns in parallel. Hence, an optimal crawling strategy is required to identify and exploit overlaps between campaigns and to merge the corresponding API requests.

Temporal Modeling. A second challenge is the modeling of large-scale graphs, where both nodes and edges have temporally evolving properties. Our approach is to treat such graphs as directed multigraphs, with multiple timestamped edges between two vertexes [6]. Given that the scale of the graphs can be very large both in terms of the number of vertexes and edges, but also along the temporal dimension, it is necessary to investigate efficient encoding and indexing for vertexes and edges, as well as their attributes. We have collected a set of tweets over a period of 20 days using Twitter's streaming API. In this set of tweets, we have counted a total of 2,406,250 distinct hashtags and 3,257,760 distinct pairs of co-occurring hashtags. If we aggregate co-occurring pairs of hashtags per hour, then we count a total of 5,670,528 distinct pairs of co-occurring hashtags. Note that the sample of tweets we have collected is only a small fraction of the total number of tweets posted on Twitter. If we can access a larger sample of tweets and consider not only hashtags but also plain terms, then the corresponding graphs will be substantially larger.

Advanced Querying. A third challenge is the definition of querying operators that can be efficiently applied on temporally evolving graphs. Algorithms such as PageRank, which compute the importance of nodes in a graph, would have to be computed for each snapshot of the temporally evolving graph. While there are approaches that can efficiently apply such algorithms on very large graphs [7], they do not consider the temporal aspect of the graphs that is present in our setting. Overall, the implementation of querying operators should exploit any redundancy or repetition in the temporally evolving graphs to speed-up the calculations.

Data Analysis. The proposed platform allows users to define a body of tweets based on complex conditions ("target definition" in Fig. **1**), in addition to the manual selection of keywords or hashtags. Since the quality of any analysis process is affected by the input data, a challenge that arises is the estimation of the bias of the results with respect to the input data.

5 Conclusions and Future Work

In this paper we have proposed and described in detail a platform for supporting data analytics tasks, such as opinion mining campaigns, on online social networks. The main focus of our approach is not on a specific analysis task per se, but rather on the proper selection of the data that constitute the input of the task. In our view, selecting the proper data set is paramount for the analytics task to produce credible results. For this reason we have directed our work as follows: (a) we are currently implementing a campaign manager for crawling Twitter based on highly configurable and adaptive campaign definitions, and (b) we have defined a preliminary model and query operators [6] for selecting tweets that satisfy complex conditions on the terms they contain, their associations, and their evolution and temporal characteristics. Our next steps include the specification and implementation of a query language that encompasses the query operators mentioned above, and the integration of the implemented components into the platform we described in this paper.

References

1. Y. Doytsher, B. Galon, and Y. Kanza. Querying geo-social data by bridging spatial networks and social networks. 2nd ACM SIGSPATIAL International Workshop on Location Based Social Networks, LBSN '10, pages 39-46, New York, NY, USA, 2010.
2. W. E. Moustafa, G. Namata, A. Deshpande, and L. Getoor. Declarative analysis of noisy information networks. 2011 IEEE 27th International Conference on Data Engineering Workshops, ICDEW '11, pages 106-111, Washington, DC, USA, 2011.
3. M. San Martin and C. Gutierrez. Representing, querying and transforming social networks with rdf/sparql. 6th European Semantic Web Conference on The Semantic Web: Research and Applications, ESWC 2009 Heraklion, pages 293-307.
4. M. A. Smith and V. Barash. Social sql: Tools for exploring social databases. IEEE Data Eng. Bull., 31(2):50-57, 2008.
5. M. Cheong and S. Ray. A Literature Review of Recent Microblogging Developments. Technical Report TR-2011-263, Clayton School of Information Technology, Monash University, 2011.
6. V. Plachouras and Y. Stavrakas. Querying Term Associations and their Temporal Evolution in Social Data. 1st Intl VLDB Workshop on Online Social Systems (WOSS 2012), Istanbul, August 2012.
7. U. Kang, D.H. Chau, C. Faloutsos: Mining large graphs: Algorithms, inference, and discoveries. ICDE 2011: 243-254.

SEKI@home, or Crowdsourcing an Open Knowledge Graph

Thomas Steiner[1]* and Stefan Mirea[2]

[1] Universitat Politècnica de Catalunya – Department LSI, Barcelona, Spain
tsteiner@lsi.upc.edu
[2] Computer Science, Jacobs University Bremen, Germany
s.mirea@jacobs-university.de

Abstract. In May 2012, the Web search engine Google has introduced the so-called Knowledge Graph, a graph that understands real-world entities and their relationships to one another. It currently contains more than 500 million objects, as well as more than 3.5 billion facts about and relationships between these different objects. Soon after its announcement, people started to ask for a programmatic method to access the data in the Knowledge Graph, however, as of today, Google does not provide one. With *SEKI@home*, which stands for *Search for Embedded Knowledge Items*, we propose a browser extension-based approach to crowdsource the task of populating a data store to build an Open Knowledge Graph. As people with the extension installed search on Google.com, the extension sends extracted anonymous Knowledge Graph facts from Search Engine Results Pages (SERPs) to a centralized, publicly accessible triple store, and thus over time creates a SPARQL-queryable Open Knowledge Graph. We have implemented and made available a prototype browser extension tailored to the Google Knowledge Graph, however, note that the concept of *SEKI@home* is generalizable for other knowledge bases.

1 Introduction

1.1 The Google Knowledge Graph

With the introduction of the Knowledge Graph, the search engine Google has made a significant paradigm shift towards *"things, not strings"* [7], as a post on the official Google blog states. Entities covered by the Knowledge Graph include landmarks, celebrities, cities, sports teams, buildings, movies, celestial objects, works of art, and more. The Knowledge Graph enhances Google search in three main ways: by disambiguation of search queries, by search log-based summarization of key facts, and by explorative search suggestions. This triggered demand for a method to access the facts stored in the Knowledge Graph programmatically [6]. At time of writing, however, no such programmatic method is available.

[*] Full disclosure: T. Steiner is also a Google employee, S. Mirea a Google intern.

1.2 On Crowdsourcing

The term *crowdsourcing* was first coined by Jeff Howe in an article in the magazine Wired [2]. It is a *portmanteau* of "crowd" and "outsourcing". Howe writes: *"The new pool of cheap labor: everyday people using their spare cycles to create content, solve problems, even do corporate R&D"*. The difference to outsourcing is that the crowd is undefined by design. We suggest crowdsourcing for the described task of extracting facts from SERPs with Knowledge Graph results for two reasons: *(i)* there is no publicly available list of the 500 million objects [7] in the Knowledge Graph, and *(ii)* even if there was such a list, it would not be practicable (nor allowed by the terms and conditions of Google) to crawl it.

1.3 Search Results as Social Media

Kaplan and Haenlein have defined social media as *"a group of Internet-based applications that build on the ideological and technological foundations of Web 2.0, and that allow the creation and exchange of user-generated content"* [4]. We argue that search results are social media as well, especially in the case of Google with its tight integration of Google+, a feature called *Search plus Your World* [8].

1.4 Contributions and Paper Structure

In this position paper, we describe and provide a prototype implementation of an approach, tentatively titled *SEKI@home* and based on crowdsourcing via a browser extension, to make closed knowledge bases programmatically and openly accessible. We demonstrate its applicability with the Google Knowledge Graph. The extension can be added to the Google Chrome browser by navigating to `http://bit.ly/SEKIatHome`, the Open Knowledge Graph SPARQL endpoint can be tested at `http://openknowledgegraph.org/sparql`[1].

The remainder of this paper is structured as follows. In Section 2, we highlight related work for the field of extracting data from websites with RDF wrappers. In Section 3, we describe the *SEKI@home* approach in detail. We provide a short evaluation in Section 4. The paper ends with an outlook on future work in Section 5 and a conclusion in Section 6.

2 Related Work

Wrappers around Web services or Web pages have been used in the past to lift data from the original source to a meaningful, machine-readable RDF level. Examples are the Google Art wrapper by Guéret [1], which lifts the data from the Google Art project [9], or the now discontinued SlideShare wrapper[2] by the same author. Such wrappers typically work by mimicking the URI scheme of the site they are wrapping. Adapting parts of the URL of the original resource to that of the wrapper provides access to the desired data. Wrappers do not offer SPARQL endpoints, as their data gets computed on-the-fly.

[1] The SPARQL endpoint and the extension were active from Aug. 11 to Sep. 6, 2012.

[2] `http://linkeddata.few.vu.nl/slideshare/`

With *SEKI@home*, we offer a related, however, still different in the detail, approach to lift and make machine-readably accessible closed knowledge bases like the Knowledge Graph. The entirety of the knowledge base being unknown, via crowdsourcing we can distribute the heavy burden of crawling the whole Knowledge Graph on many shoulders. Finally, by storing the extracted facts centrally in a triple store, our approach allows for openly accessing the data via the standard SPARQL protocol.

3 Methodology

3.1 Browser Extensions

We have implemented our prototype browser extension for the Google Chrome browser. Chrome extensions are small software programs that users can install to enrich their browsing experience. Via so-called *content scripts*, extensions can inject and modify the contents of Web pages. We have implemented an extension that gets activated when a user uses Google to search the Web.

3.2 Web Scraping

Web scraping is a technique to extract data from Web pages. We use CSS selectors [3] to retrieve page content from SERPs that have an associated real-world entity in the Knowledge Graph. An exemplary query selector is `.kno-desc` (all elements with class name “kno-desc”), which via the JavaScript command `document.querySelector` returns the description of a Knowledge Graph entity.

3.3 Lifting the Extracted Knowledge Graph Data

Albeit the claim of the Knowledge Graph is *“things, not strings”* [7], what gets displayed to search engine users are strings, as can be seen in a screenshot available at `http://twitpic.com/ahqqls/full`. In order to make this data meaningful again, we need to lift it. We use JSON-LD [10], a JSON representation format for expressing directed graphs; mixing both Linked Data and non-Linked Data in a single document. JSON-LD allows for adding meaning by simply including or referencing a so-called (data) context. The syntax is designed to not disturb already deployed systems running on JSON, but to provide a smooth upgrade path from JSON to JSON-LD.

We have modeled the plaintext Knowledge Graph terms (or predicates) like “Born”, “Full name”, “Height”, “Spouse”, etc. in an informal Knowledge Graph ontology under the namespace `okg` (for Open Knowledge Graph) with spaces converted to underscores. This ontology has already been partially mapped to common Linked Data vocabularies. One example is `okg:Description`, which directly maps to `dbpprop:shortDescription` from DBpedia. Similar to the unknown list of objects in the Knowledge Graph (see Subsection 1.2), there is no known list of Knowledge Graph terms, which makes a complete mapping impossible. We have collected roughly 380 Knowledge Graph terms at time of writing, however, mapping them to other Linked Data vocabularies will be a permanent work in progress. As an example, Listing 1 shows the lifted, meaningful JSON-LD as returned by the extension.

```
{
  "@id": "http://openknowledgegraph.org/data/H4sIAAAAA[...]",
  "@context": {
    "Name": "http://xmlns.com/foaf/0.1/name",
    "Topic_Of": {
      "@id": "http://xmlns.com/foaf/0.1/isPrimaryTopicOf",
      "type": "@id"
    },
    "Derived_From": {
      "@id": "http://www.w3.org/ns/prov#wasDerivedFrom",
      "type": "@id"
    },
    "Fact": "http://openknowledegraph.org/ontology/Fact",
    "Query": "http://openknowledegraph.org/ontology/Query",
    "Full_name": "http://xmlns.com/foaf/0.1/givenName",
    "Height": "http://dbpedia.org/ontology/height",
    "Spouse": "http://dbpedia.org/ontology/spouse"
  },
  "Derived_From": "http://www.google.com/insidesearch/
      features/search/knowledge.html",
  "Topic_Of": "http://en.wikipedia.org/wiki/Chuck_Norris",
  "Name": "Chuck Norris",
  "Fact": ["Chuck Norris can cut thru a knife w/ butter."],
  "Full_name": ["Carlos Ray Norris"],
  "Height": ["5' 10\""],
  "Spouse": [
    {
      "@id": "http://openknowledgegraph.org/data/H4sIA[...]",
      "Query": "gena o'kelley",
      "Name": "Gena O'Kelley"
    },
    {
      "@id": "http://openknowledgegraph.org/data/H4sIA[...]",
      "Query": "dianne holechek",
      "Name": "Dianne Holechek"
    }
  ]
}
```

Listing 1. Subset of the meaningful JSON-LD from the Chuck Norris Knowledge Graph data. The mapping of the Knowledge Graph terms can be seen in the @context.

3.4 Maintaining Provenance Data

The facts extracted via the *SEKI@home* approach are derived from existing third-party knowledge bases, like the Knowledge Graph. A derivation is a transformation of an entity into another, a construction of an entity into another, or an update of an entity, resulting in a new one. In consequence, it is considered good form to acknowledge the original source, *i.e.*, the Knowledge Graph, which we have done via the property `prov:wasDerivedFrom` from the PROV Ontology [5] for each entity.

4 Evaluation

4.1 Ease of Use

At time of writing, we have evaluated the *SEKI@home* approach for the criterium *ease of use* with a number of 15 users with medium to advanced computer and programming skills who had installed a pre-release version of the browser extension and who simply browsed the Google Knowledge Graph by following links, starting from the URL `https://www.google.com/search?q=chuck+norris`, which triggers Knowledge Graph results. One of our design goals when we imagined *SEKI@home* was to make it as unobtrusive as possible. We asked the extension users to install the extension and tell us if they noticed any difference at all when using Google. None of them noticed any difference, while actually in the background the extension was sending back extracted Knowledge Graph facts to the RDF triple store at full pace.

4.2 Data Statistics

On average, the number of 31 triples gets added to the triple store per SERP with Knowledge Graph result. Knowledge Graph results vary in their level of detail. We have calculated an average number of about 5 Knowledge Graph terms (or predicates) per SERP with Knowledge Graph result. While some Knowledge Graph values (or objects) are plaintext strings like the value "Carlos Ray Norris" for `okg:Full_name`, others are references to other Knowledge Graph entities, like a value for `okg:Movies_and_TV_shows`. The relation of reference values to plaintext values is about 1.5, which means the Knowledge Graph is well interconnected.

4.3 Quantitative Evaluation

In its short lifetime from August 11 to September 6, 2012, the extension users have collected exactly 2,850,510 RDF triples. In that period, all in all 39 users had the extension installed in production.

5 Future Work

A concrete next step for the current application of our approach to the Knowledge Graph is to provide a more comprehensive mapping of Knowledge Graph terms to other Linked Data vocabularies, a task whose difficulty was outlined in Subsection 3.3. At time of writing, we have applied the *SEKI@home* approach to a concrete knowledge base, namely the Knowledge Graph. In the future, we want to apply *SEKI@home* to similar closed knowledge bases. Videos from video portals like YouTube or Vimeo can be semantically enriched, as we have shown in [11] for the case of YouTube. We plan to apply *SEKI@home* to semantic video enrichment by splitting the computational heavy annotation task, and store the extracted facts centrally in a triple store to allow for open SPARQL access. In [12], we have proposed the creation of a comments archive of things people said about real-world entities on social networks like Twitter, Facebook, and Google+, which we plan to realize via *SEKI@home*.

6 Conclusion

In this paper, we have shown a generalizable approach to first open up closed knowledge bases by means of crowdsourcing, and then make the extracted facts universally and openly accessible. As an example knowledge base, we have used the Google Knowledge Graph. The extracted facts can be accessed via the standard SPARQL protocol from the Google-independent Open Knowledge Graph website (`http://openknowledgegraph.org/sparql`). Just like knowledge bases evolve over time, the Knowledge Graph in concrete, the facts extracted via the *SEKI@home* approach as well mirror those changes eventually. Granted that provenance of the extracted data is handled appropriately, we hope to have contributed a useful socially enabled chain link to the Linked Data world.

Acknowledgments

T. Steiner is partially supported by the EC under Grant No. 248296 FP7 (I-SEARCH).

References

1. C. Guéret. "GoogleArt — Semantic Data Wrapper (Technical Update)", SemanticWeb.com, Mar. 2011. `http://semanticweb.com/googleart-semantic-data-wrapper-technical-update_b18726`.
2. J. Howe. The Rise of Crowdsourcing. *Wired*, 14(6), June 2006. `http://www.wired.com/wired/archive/14.06/crowds.html`.
3. L. Hunt and A. van Kesteren. Selectors API Level 1. Candidate Recommendation, W3C, June 2012. `http://www.w3.org/TR/selectors-api/`.
4. A. M. Kaplan and M. Haenlein. Users of the world, unite! The challenges and opportunities of Social Media. *Business Horizons*, 53(1):59–68, Jan. 2010.
5. T. Lebo, S. Sahoo, D. McGuinness, K. Belhajjame, J. Cheney, D. Corsar, D. Garijo, S. Soiland-Reyes, S. Zednik, and J. Zhao. PROV-O: The PROV Ontology. Working Draft, W3C, July 2012. `http://www.w3.org/TR/prov-o/`.
6. Questioner on Quora.com. "Is there a Google Knowledge Graph API (or another third party API) to get semantic topic suggestions for a text query?", May 2012. `http://bit.ly/Is-there-a-Google-Knowledge-Graph-API`.
7. A. Singhal. "Introducing the Knowledge Graph: things, not strings", Google Blog, May 2012. `http://googleblog.blogspot.com/2012/05/introducing-knowledge-graph-things-not.html`.
8. A. Singhal. "Search, plus Your World", Google Blog, Jan. 2012. `http://googleblog.blogspot.com/2012/01/search-plus-your-world.html`.
9. A. Sood. "Explore museums and great works of art in the Google Art Project", Google Blog, Feb. 2011. `http://googleblog.blogspot.com/2011/02/explore-museums-and-great-works-of-art.html`.
10. M. Sporny, D. Longley, G. Kellogg, M. Lanthaler, and M. Birbeck. JSON-LD Syntax 1.0, A Context-based JSON Serialization for Linking Data. Working Draft, W3C, July 2012. `http://www.w3.org/TR/json-ld-syntax/`.
11. T. Steiner. SemWebVid – Making Video a First Class Semantic Web Citizen and a First Class Web Bourgeois. In *Proceedings of the ISWC 2010 Posters & Demonstrations Track*, Nov. 2010.
12. T. Steiner, R. Verborgh, R. Troncy, J. Gabarro, and R. V. de Walle. Adding Realtime Coverage to the Google Knowledge Graph. In *Proceedings of the ISWC 2012 Posters & Demonstrations Track*. (accepted for publication).

Cluster-based Instance Consolidation For Subsequent Matching

Jennifer Sleeman and Tim Finin
Department of Computer Science and Electrical Engineering
University of Maryland Baltimore County
Baltimore. MD 21250 USA jsleem1,finin@cs.umbc.edu

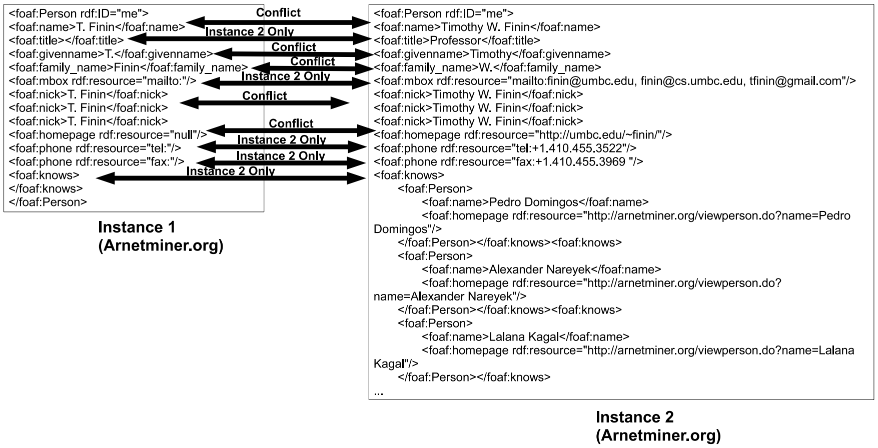
September 7, 2012

Abstract

Instance consolidation is a way to merge instances that are thought to be the same or closely related that can be used to support coreference resolution and entity linking. For Semantic Web data, consolidating instances can be as simple as relating instances using owl:sameAs, as is the case in linked data, or merging instances that could then be used to populate or enrich a knowledge model. In many applications, systems process data incrementally over time and as new data is processed, the state of the knowledge model changes. Previous consolidations could prove to be incorrect. Consequently, a more abstract representation is needed to support instance consolidation. We describe our current research to perform consolidation that includes temporal support, support to resolve conflicts and an abstract representation of an instance that is the aggregate of a cluster of matched instances. We believe that this model will prove flexible enough to handle sparse instance data and can improve the accuracy of the knowledge model over time.

1 Introduction

Though consolidation has been researched in other domains, such as the database domain, it is less explored in the Semantic Web domain. In relation to coreference resolution (also known as instance matching and entity resolution), once two instances or entities are designated as the same or coreferent, they are tagged in some way (using owl:sameAs) or consolidated into a single entity using various approaches. What has received less attention is how to merge instances with conflicted information and how to adapt consolidations over time. In this paper we describe our ongoing work that supports instance consolidation by grouping matched instances into clusters of abstract representations. We develop our consolidation algorithm to work with incremental online coreference resolution by providing a way to improve the instance data that will be used in subsequent matching. For example, in the case of sparse instances, a consolidated representation of features would be more likely to match newly discovered instances. As more instances are added to the cluster, the representation will become more enriched and more likely



```
<foaf:Person rdf:ID="me">
<foaf:name>T. Finin</foaf:name>
<foaf:title></foaf:title>
<foaf:givenname>T.</foaf:givenname>
<foaf:family_name>Finin</foaf:family_name>
<foaf:mbox rdf:resource="mailto:"/>
<foaf:nick>T. Finin</foaf:nick>
<foaf:nick>T. Finin</foaf:nick>
<foaf:nick>T. Finin</foaf:nick>
<foaf:homepage rdf:resource="null"/>
<foaf:phone rdf:resource="tel:"/>
<foaf:phone rdf:resource="fax:"/>
<foaf:knows>
</foaf:knows>
</foaf:Person>
```

```
<foaf:Person rdf:ID="me">
<foaf:name>Timothy W. Finin</foaf:name>
<foaf:title>Professor</foaf:title>
<foaf:givenname>Timothy</foaf:givenname>
<foaf:family_name>W.</foaf:family_name>
<foaf:mbox rdf:resource="mailto:finin@umbc.edu, finin@cs.umbc.edu, tfinin@gmail.com"/>
<foaf:nick>Timothy W. Finin</foaf:nick>
<foaf:nick>Timothy W. Finin</foaf:nick>
<foaf:nick>Timothy W. Finin</foaf:nick>
<foaf:homepage rdf:resource="http://umbc.edu/~finin/"/>
<foaf:phone rdf:resource="tel:+1.410.455.3522"/>
<foaf:phone rdf:resource="fax:+1.410.455.3969 "/>
<foaf:knows>
    <foaf:Person>
        <foaf:name>Pedro Domingos</foaf:name>
        <foaf:homepage rdf:resource="http://arnetminer.org/viewperson.do?name=Pedro
Domingos"/>
    </foaf:Person></foaf:knows><foaf:knows>
    <foaf:Person>
        <foaf:name>Alexander Nareyek</foaf:name>
        <foaf:homepage rdf:resource="http://arnetminer.org/viewperson.do?
name=Alexander Nareyek"/>
    </foaf:Person></foaf:knows><foaf:knows>
    <foaf:Person>
        <foaf:name>Lalana Kagal</foaf:name>
        <foaf:homepage rdf:resource="http://arnetminer.org/viewperson.do?name=Lalana
Kagal"/>
    </foaf:Person></foaf:knows>
...
```

Figure 1: Conflicts During A Merge

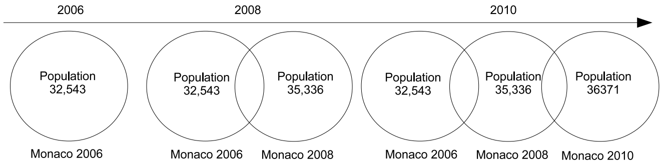
to match a wider number of instances in subsequent matches. When performing subsequent instance matching that includes both clusters and individual instances, the consolidated representation of clustered data, supported by our merging algorithm, can be used.

Figure 1 depicts an example of a consolidation when conflicts may occur. In this example, when we have a pair of attributes that are the same but their values differ, to consolidate we must determine whether both values are maintained, none of the values are maintained or one of the values is maintained. For the purposes of using the consolidated instance for future matching, the merging of instance data is incredibly important as it affects the performance of future matching. This is particularly true when working with data sets that are sparse.

The temporal support is an important aspect to this problem since over time an entity's features may change. In Figure 2, the attribute population changes over time. This example highlights two complexities that are a natural effect of time. An instance can be thought of as a snapshot in time, therefore an instance captured at time $t-1$ may not be as relevant as an instance captured at time t. This affects how instances should be consolidated and is a good example of when a technique is required to resolve conflicts. Also, this implies that in certain cases, given enough time, two instances may no longer be coreferent, supporting the argument that temporal issues play a significant role in consolidation and subsequent processing.

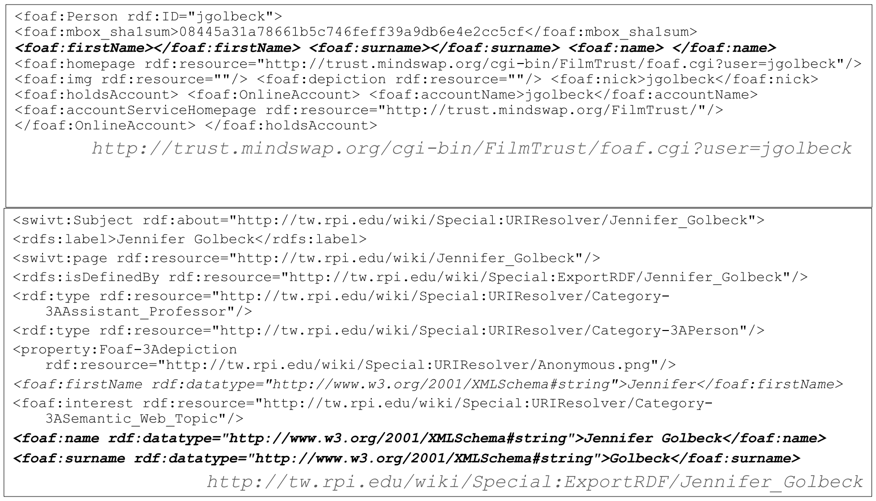
2 Background

Semantic Web data, which includes semantically tagged data represented using a Resource Description Framework (RDF) [1, 2] triples (subject, predicate, object) format, is often used as a way to commonly represent data. Data which conforms to an ontology, data exported from social networking sites, and linked data found on the Linked Open Data Cloud are often represented as triples. Attempting to match instances or entities among this type of data can be a challenge which is further complicated by noise and data spareness.



Sources:
http://www.infoplease.com/country/profiles/monaco.html
http://www.tradingeconomics.com/monaco/population-total-wb-data.html
http://en.wikipedia.org/wiki/Monaco

Figure 2: Temporal Aspects of Consolidation



```
<foaf:Person rdf:ID="jgolbeck">
<foaf:mbox_sha1sum>08445a31a78661b5c746feff39a9db6e4e2cc5cf</foaf:mbox_sha1sum>
<foaf:firstName></foaf:firstName> <foaf:surname></foaf:surname> <foaf:name> </foaf:name>
<foaf:homepage rdf:resource="http://trust.mindswap.org/cgi-bin/FilmTrust/foaf.cgi?user=jgolbeck"/>
<foaf:img rdf:resource=""/> <foaf:depiction rdf:resource=""/> <foaf:nick>jgolbeck</foaf:nick>
<foaf:holdsAccount> <foaf:OnlineAccount> <foaf:accountName>jgolbeck</foaf:accountName>
<foaf:accountServiceHomepage rdf:resource="http://trust.mindswap.org/FilmTrust/"/>
</foaf:OnlineAccount> </foaf:holdsAccount>
         http://trust.mindswap.org/cgi-bin/FilmTrust/foaf.cgi?user=jgolbeck
```

```
<swivt:Subject rdf:about="http://tw.rpi.edu/wiki/Special:URIResolver/Jennifer_Golbeck">
<rdfs:label>Jennifer Golbeck</rdfs:label>
<swivt:page rdf:resource="http://tw.rpi.edu/wiki/Jennifer_Golbeck"/>
<rdfs:isDefinedBy rdf:resource="http://tw.rpi.edu/wiki/Special:ExportRDF/Jennifer_Golbeck"/>
<rdf:type rdf:resource="http://tw.rpi.edu/wiki/Special:URIResolver/Category-
    3AAssistant_Professor"/>
<rdf:type rdf:resource="http://tw.rpi.edu/wiki/Special:URIResolver/Category-3APerson"/>
<property:Foaf-3Adepiction
    rdf:resource="http://tw.rpi.edu/wiki/Special:URIResolver/Anonymous.png"/>
<foaf:firstName rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Jennifer</foaf:firstName>
<foaf:interest rdf:resource="http://tw.rpi.edu/wiki/Special:URIResolver/Category-
    3ASemantic_Web_Topic"/>
<foaf:name rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Jennifer Golbeck</foaf:name>
<foaf:surname rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Golbeck</foaf:surname>
                         http://tw.rpi.edu/wiki/Special:ExportRDF/Jennifer_Golbeck
```

Figure 3: Consolidating Instances

The topic of instance consolidation, the process of combining instances, is not novel. Previous research has addressed instance consolidation in relation to merging instances that are coreferent. What has received less attention is the temporal aspect of this problem, how to merge instances when conflicts are present and how using this method to support incremental coreference resolution can address issues related to spareness. For example, in Figure 3 we show two Friend of a Friend (FOAF) [3] documents representing a person. In the top document, information such as foaf:firstName, foaf:surname and foaf:name is absent. In the bottom document, these values exist and so a consolidation of these two documents would eliminate attributes that are missing values and increase the number of features that could be matched for subsequent matching.

The research that exists today, has a tendency to use a methodology that relies upon inverse functional properties. For example, Hogan et al. [4] use inverse functional properties to determine instances in common and rewrite identifiers based on each equivalence chain. They require retrieval of the ontologies to identify inverse functional properties, this is not always possible. They describe a merge process that assumes agreement, i.e. no conflicts and they do not address how to handle

Source	Avg Number of Attributes	Number of Instances
vox	5.65	4492
journal	9.71	1259
ebiquity	19.78	217

Table 1: Average Number of Attributes

data that does not use inverse functional properties. Shi et al. [5] describe instance consolidation as 'smushing' and performs 'smushing' by taking advantage of the inverse functional property. They work at the attribute level and calculate attribute level similarity measures. A property defined as inverse functional implies that the inverse of the property is functional; that it uniquely identifies the subject [2]. Again this work relies upon inverse functional properties and tends not to address how to resolve conflicts. Yatskevich et al. [6] address consolidation of graphs. They merge graphs if the instances belong to the same class, and if their string similarity is higher than a threshold. They describe special cases for particular types. This merge process does not address conflicts and there is no indication whether they could reverse a consolidated graph. In our previous work [7, 8] that explored our approach using simple merging heuristics and coreferent clustering of FOAF instances, particularly when working with sparse input, consolidation did positively affect subsequent coreferent pairing.

In our person data set, specifically using the FOAF ontology, we found a sizable percentage of the instances contained very few attributes. In Table 1, we show the number of instances originating from 3 different sources. Source 'vox' had the highest number of instances and also the lowest number of attributes per instance. We have found this is prevalent among social networking sites and sites that support exports of user profile data using the FOAF ontology. This is not specific to FOAF instances and can present a problem for coreference resolution algorithms.

3 An Approach

We define an instance as an abstract representation that can be either a cluster of coreferent instances, or a single entity instance. A formal definition follows.

Definition 1. *Given a set of instances I and a set of clusters C, an abstract instance $A \in (I \cup C)$.*

Definition 2. *Given a pair of instances i_n and i_m, if the pair are coreferent or $coref\,(i_n, i_m)$, then a cluster C_{nm} is formed such that the cluster $C_{nm} = \{i_n, i_m\}$.*

Figure 4 depicts an example of a cluster that is formed with coreferent instances. Data relates to Monaco from three different sources (http://dbpedia.org/, http://www4.wiwiss.fu-berlin.de/, http://data.nytimes.com/) where each source represents a perspective of Monaco. Given a system that processes unstructured text that includes a reference to the instance Monaco, our work seeks to prove that we are more likely to recognize Monaco as an entity with the combined information taking the most relevant of features, rather than using a single instance. We will also show how over time as attributes pertaining to these instances changes, the model can reflect these changes.

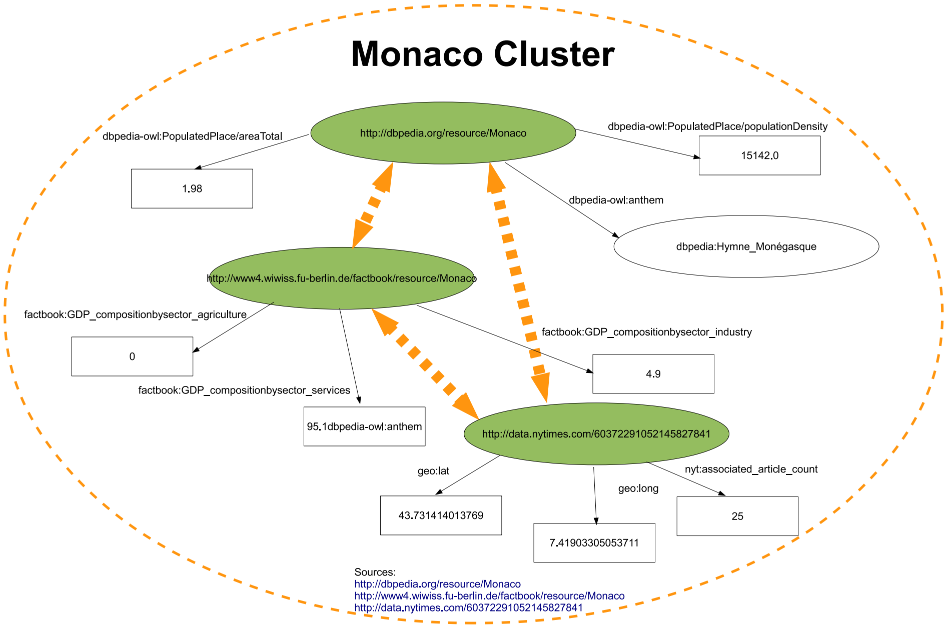

Monaco Cluster
http://dbpedia.org/resource/Monaco
dbpedia-owl:PopulatedPlace/areaTotal
1.98
dbpedia-owl:PopulatedPlace/populationDensity
15142.0
dbpedia-owl:anthem
dbpedia:Hymne_Monégasque
http://www4.wiwiss.fu-berlin.de/factbook/resource/Monaco
factbook:GDP_compositionbysector_agriculture
0
factbook:GDP_compositionbysector_industry
4.9
factbook:GDP_compositionbysector_services
95.1dbpedia-owl:anthem
http://data.nytimes.com/60372291052145827841
geo:lat
43.731414013769
geo:long
7.41903305053711
nyt:associated_article_count
25
Sources:
http://dbpedia.org/resource/Monaco
http://www4.wiwiss.fu-berlin.de/factbook/resource/Monaco
http://data.nytimes.com/60372291052145827841

Figure 4: Instance Cluster

A consolidated representation is required in order to use the clustered data in subsequent matching. This consolidation can be as simple as a union between the sets of attributes of instances. However, as seen in Figure 1, this approach does not address situations where attributes are in conflict. Even in this simple example, conflicts are present that should be resolved. Our initial work includes the merging of instances and resolution of conflicts by using a basic set of rules and temporal information.

When evaluating two instances, for each attribute that is shared, if the values are equal, we retain one instance of the attribute. If two instances share the same attribute but their values differ, we try to resolve the conflict. If the two instances contain attributes that are not shared we include their unshared attributes. In resolving the conflict, we first try to determine if the two values are synonymous. If they are synonymous, we keep both values for the particular attribute. If not, then we will use additional analysis such as temporal information. As the same instance is processed over time given a particular URI, we track the changes among attributes for that instance. Given that attributes have changed for a particular instance we give the more recent values of the attributes a higher significance than a less recent values. We can then use this information to assist with resolving conflicts. When conflicts can not be resolved we keep both values for the unresolved attribute. We anticipate this approach will advance as we progress in our research.

Our cluster links are symbolic in nature. In order to support changes to the cluster over time, each instance in a cluster is linked and weighted to other instances in the cluster. How the weight is defined is based on the coreference resolution algorithm. In our work, we are using a clustering algorithm to cluster instances that are thought to be coreferent. The output of our calculation that supports our clustering method will also be used as an assessment of how closely the two instances are related for consolidation. Given the set of attributes for each instance in the cluster, we associate a score with each set of matched attributes. This score can be based on a distance function or based on a more complex representation. The goal of this step is to weight common features among pairs of coreferent instances in the cluster. Across all features in the cluster we wish to pick the most significant features to be used for subsequent matching. We are currently exploring feature reduction mechanisms to perform this step. This structure gives us the ability to compare coreferent relationships among instances over time, to remove coreferent relationships given changes over time, or to add and modify existing relationships given new instances that are added to the cluster.

4 Conclusion

We have presented a need for a more adaptive-based consolidation approach. A cluster-based consolidation provides a powerful model for instance matching algorithms. It is meant to adapt to change over time, is flexible and could potentially improve subsequent matching. Given the complexities of systems today, adaptation is a necessity. The challenge is developing a consolidation approach that is flexible enough to support the complexities of systems today, without incurring a large performance penalty.

References

[1] Beckett, D.: Rdf/xml syntax specification. http://www.w3.org/TR/REC-rdf-syntax/ (2004)

[2] Brickley, D., Guha, R.: Resource description framework (rdf) schema specification 1.0. http://www.w3.org/TR/rdf-schema/ (2004)

[3] Brickley, D., Miller, L.: Foaf vocabulary specification .98 (August 2010) http://xmlns.com/foaf/spec/.

[4] Hogan, A., Harth, A., Decker, S.: Performing object consolidation on the semantic web data graph. In: Proc. I3: Identity, Identifiers, Identification. Workshop at 16th Int. World Wide Web Conf. (February 2007)

[5] Shi, L., Berrueta, D., Fernandez, S., Polo, L., Fernandez, S.: Smushing rdf instances: are alice and bob the same open source developer? In: Proc. 3rd Expert Finder workshop on Personal Identification and Collaborations: Knowledge Mediation and Extraction, 7th Int. Semantic Web Conf. (November 2008)

[6] Yatskevich, M., Welty, C., Murdock, J.: Coreference resolution on rdf graphs generated from information extraction: first results. In: the ISWC 06 Workshop on Web Content Mining with Human Language Technologies. (2006)

[7] Sleeman, J., Finin, T.: A machine learning approach to linking foaf instances. In: Spring Symposium on Linked Data Meets AI, AAAI (January 2010)

[8] Sleeman, J., Finin, T.: Computing foaf co-reference relations with rules and machine learning. In: The Third International Workshop on Social Data on the Web, ISWC (November 2010)

Semantically Tagging Images of Landmarks

Heather S. Packer, Jonathon S. Hare, Sina Samangooei, and Paul H. Lewis

Electronics and Computer Science,
University of Southampton,
Southampton SO17 1BJ, UK
{hp3|jsh2|ss|phl}@ecs.soton.ac.uk

Abstract. Semantic tagging allows images to be linked with URIs from web resources. Disambiguating URIs which correspond with an image's visual content is challenging. Previous work has largely failed to effectively contextualise the knowledge provided by the Semantic Web and the user-provided keyword tags in images. We propose an algorithm which uses geographical coordinates and keywords of similar images to recommend semantic tags that describe an image's visual content. Our algorithm uses the semantic stores YAGO2, DBPedia and Geonames. These stores allow us to handle multi-lingual keyword tags and disambiguate between alternative names for landmarks.

1 Introduction

Image tagging systems predominately allow no explicit semantic links describing the visual content of images. The meaning of user assigned keyword tags can be ambiguous because they typically use only a few words. For example, a user may tag an image with "cathedral". This keyword tag does not specify which cathedral. In order to improve the semantics of tags, "Semantic tags" (also known as "Machine tags") have been used to add explicit context in the form of links and additional information such as geographic coordinates. Specifically, a semantic tag is a link defined by a Uniform Resource Identifier (URI) referencing an entity defined on the Semantic Web. Semantic tags enrich images by linking them to resources that provide more information about the entities contained within the image. A key challenge in semantic tagging is assigning the correct URIs.

This paper describes an approach that attempts to semantically enrich a query image, that has no information other than its pixel content, with URIs of the entities depicted in the image. Firstly, we identify visually similar images by comparing the visual features of the query image against a large corpus of partially annotated images. This corpus contains a mixture of images that have been geotagged and images with keyword tags, or images with a mixture of the two types of annotation. Using the geolocation information from the similar images we estimate the most likely geographic coordinates of the query. We then compare the raw keyword tags of the similar images to entities contained in the YAGO2 [1] knowledge-base and validate whether any geographic-relations of the entities are close to the query image's estimated location using the DBPedia [2] and Geonames [3] knowledge-bases. Finally, we use these selected entities to construct a list of URIs which are relevant to the content of the query image[1].

The broader motivation of this work is to bridge the gap between unstructured data attainable in real time, such as a GPS-location or image of a landmark, and, rich, detailed and structured information about that landmark, therefore facilitate more informed search and retrieval. For example, users can engage in semantic searching; targeting a particular church, architect, period in history or architectural style. This kind of structured search can also support event detection [4] or lifelogging [5] by helping collate documents which refer to a particular entity more accurately.

This work provides two contributions beyond state-of the-art. Specifically, we verify semantic tags using distance according to the height of the entities identified in an image, and we also recommend URIs using a multi-language tag index derived from multiple semantic web knowledge sources. The multi-lingual index allows us to make recommendations from keywords in foreign languages by identifying alternative names to utilise in our approach.

The structure of the paper is as follows: firstly, we discuss related work on tagging systems. Secondly we present details of our tagging approach. Finally, we discuss the use of our system with some examples and provide our conclusions.

2 Related Work

Automatic image annotation is widely studied, but techniques that integrate multiple contexts using semantic web resources are relatively rare. The following review looks at works that recommend tags using both geographical coordinates and visual image similarity. SpiritTagger [6] recommends keywords that reflect the *spirit* of a location. It finds visually similar images using colour, texture, edge and SIFT features [7], and clusters tags based within a geographical radius. These selected tags are ranked based on frequency and then importance. This enables their algorithm to recommend tags that are specific to a geographic region. The focus of our work is to recommend semantic tags (URIs) that describe a place of interest, not tags relating to a larger area.

Moxley et al. [8] use a corpus of images and their tags, and organise them into sets of places, events and visual components. These are clustered based on the co-occurrence of words and the distance between named geographical entities. If an image matches, the wikipedia title is used as a recommendation for the name of the landmark in the image. This approach uses limited information from wikipedia to identify and recommend tags. In our approach, we aim to validate semantic tags using additional information from semantic data sources.

Similar to Moxley et al. [6], Kennedy and Naaman [9]'s approach also considers the importance of tags relevant to a specific area or event. Their approach generates a representative image set for landmarks using image tags and geographic coordinates. Their technique identifies tags that occur frequently within

[1] A demonstration of our approach can be found here: `http://gtb.imageterrier.org`

one geographic area, but infrequently elsewhere, in order to identify tags that are uniquely local. They also filter tags that only occur during specific time ranges; this enables them to identify events such as the "New York Marathon" and determine whether this is relevant to a query image by analysing the date it was taken. Their focus was not concerned with recommending semantic tags.

There are a number of datasets that contain flickr images and related URIs. For instance, the Ookaboo dataset was manually created by 170,000 contributors who submitted images and classify them against a topic from wikipedia [2]. In contrast, our approach recommends URIs automatically for an untagged image, by using tags from images that share visual features. The flickr$^{\text{tm}}$ wrapper API allows users to search with a URI of an entity on Wikipedia and search for images that depict that entity. In particular, you can search for images within a user-specified distance of the geographical location of the searched entity (if the entity has a geographical location). This is the opposite problem to us, our query is an image depicting a landmark where the landmark is unknown, whereas their query is an entity on Wikipedia. The work by [10] identify entities using natural language processing by stemming words to find their root or base by removing any inflections, using Wordnet. They then identify any relationships between these entities using the hypernym, holonym, meronym, and toponym relationships described in Wordnet to create the triples describing the entities described in flickr tags. Our approach supports [10]'s, by generating the URI describing entities depicted in an image when it has no tags, so that their approach could generate triples.

A number of other approaches simply use location and visual features to annotate images (e.g. [11]). There has also been work to recommend tags, based on existing annotations (e.g. [12]), and recommending semantic entities, based on existing tags (e.g. [13]).

3 Approach

Our semantic tag recommendation approach has five steps. Firstly, we search a large index of visual features extracted from partially annotated images with the features extracted from a query image in order to find images similar to the query. The index contains images that have either geographic tags or keyword tags (or a mixture of the two). From the set of similar images with geographic locations we calculate a robust average of the latitude and longitude which estimates the geographic location of the query. Secondly, we use the keyword tags associated with the similar images to identify entities close to the estimated coordinates from YAGO2. Thirdly, we classify the types of entities that are possible recommendations using the type hierarchies of YAGO2, DBPedia and Geonames. In the fourth step, we restrict our recommendations based on their height and distance. In the final step, we expand our set of URIs with those from the closest city in order to try and identify additional relevant semantic entities.

[2] Ookaboo: `http://ookaboo.com/o/pictures/`

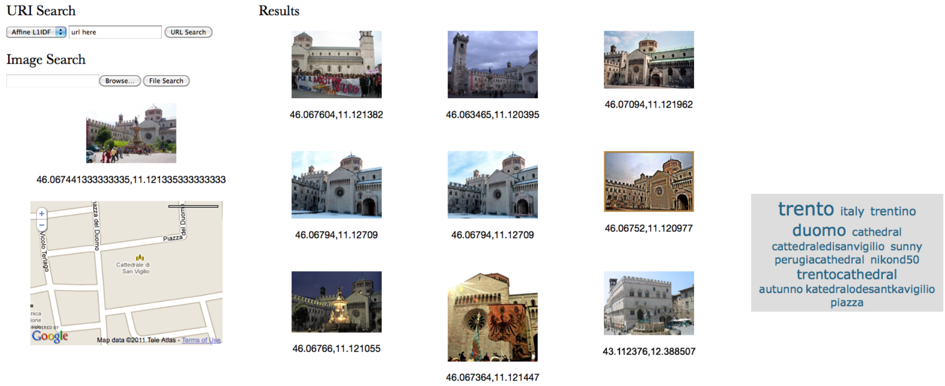


Fig. 1. The query image of Trento Cathedral, and the resulting images that matched, based on an index of SIFT features.

The approach has been developed using an index of images crawled from Flickr representing the Trentino region in Italy. More details of the dataset can be found in Section 4. In the remainder of this section, we walk through each of the five steps in detail.

3.1 Visual Image Similarity

Firstly, we compare the visual content of an untagged query with the visual content of each image in a dataset of tagged images. This is achieved by comparing via a BoVW (Bag of Visual Words) [14] representation of both query image and dataset images, extracted using the OpenIMAJ Toolkit[3] [15]. For efficient retrieval, the BoVW of the dataset images is held in a compressed inverted index, constructed using ImageTerrier[4] [16]. Once constructed, the index can be used to retrieve dataset images which are most visually similar to a query image; the tags and geographic locations of the closest dataset images are passed on to the next steps of our process. Specifically, the retrieval engine is tuned to only retrieve images that match with a very high confidence and thus only match the specific landmark/object in the query image; the aim is not to classify images into landmark classes, but rather to identify a specific instance.

The BoVW image representations are constructed by extracting difference-of-Gaussian SIFT features [17] from an of image and quantising them to a discrete vocabulary. A vocabulary of 1 million features was learnt using approximate K-Means clustering [18] with SIFT features from the MIRFlickr25000 dataset [19]. Once the content of each image is represented as a set of visual terms, we construct an inverted index which encodes each term in an image. The inverted index is augmented with the orientation information of the SIFT feature corresponding to each term; this extra geometric information allows us to improve

[3] OpenIMAJ Toolkit: `http://openimaj.org`

[4] ImageTerrier: `http://imageterrier.org`

retrieval precision using an orientation consistency check [20] at query time. For every query image, the SIFT features are extracted and quantised. The set of visual terms form a query against the inverted index which is evaluated using the Inverse-Document-Frequency weighted L1 distance metric [21]. Using this strategy we select the top ten images in the dataset. These images provide us with potential keyword tags and geographic coordinates.

An iterative approach is used to robustly estimate the geographic location of the query from the set of geotagged result images. The technique needs to be robust as there is a high probability of outliers. Starting with all the matching geotagged images, a geographic centroid is found and the image which is geographically furthest from the centre is removed. The centroid is then updated with the remaining images. This process continues iteratively until the distance between the current centroid and furthest point from the centroid is less than a predefined threshold. Through initial tests on our dataset, we found that the threshold of 0.8 returned between 70% and 100% of images that were visually similar. An example of the visual similarity search and geographic localisation for a query image of Trento Cathedral is illustrated in Figure 1.

3.2 Keyword-Entity Matching

The second step aims to find URIs representing entities in the query image by attempting to match keyword tags to the names of entities. This can be problematic because it is common for keyword tags to contain more than one word without white space. Therefore, when searching for an entity in YAGO2 that matches a tag representing more than one word it will yield no matches. For example, the keyword tag 'trentocathedral' will not match the YAGO2 entity 'Trento Cathedral'. In order to enable us to search for flattened tags, we performed a pre-processing step to create additional triples relating flattened tags to entities within YAGO2. We also flattened the entities relating to an entity through the "isCalled" property because it contains alternate terms used to refer to an instance (including foreign language names). For example, the YAGO2 entity for "Trento Cathedral" can also be called "Cattedrale di San Vigilio" and "Katedralo de Santka Vigilio". Thus, we also use the flattened entity names "cattedraledisanvigilio" and "katedralodesantkavigilio" to represent "Trento Cathedral". These additional triples and YAGO2 are used to look up all the tags using exact string matching. If there are matching entities then we check that they are in the same city (using the geographical coordinates from YAGO2 and the estimated coordinates from step one). In our Trento example, we retrieve the URIs shown in Table 1 from the image's tags.

3.3 Cross-Source Category Matches

The aim of the third step is to determine whether the entities identified in step 2 and the keyword tags of the similar images are of a specific type. We organised these types into general categories, including town/city, region, country, date, weather, season, mountain, building, activity, transport and unknown. This list

Table 1. The tags and YAGO2 matches for the first four result images.

Image	Tags	YAGO2 Matches

1	cathedral, trento, duomo	`http://mpii.de/yago/resource/Trento_Cathedral`
		`http://mpii.de/yago/resource/Cathedral`
		`http://mpii.de/yago/resource/Trento`

2	trento, italy, trentino, duomo	`http://mpii.de/yago/resource/Province_of_Trento`
		`http://mpii.de/yago/resource/Cathedral`
		`http://mpii.de/yago/resource/Trento`
		`http://mpii.de/yago/resource/Italy`

3	cattedrale-disanvigilio, cathedral, trento	`http://mpii.de/yago/resource/Trento_Cathedral`
		`http://mpii.de/yago/resource/Cathedral`
		`http://mpii.de/yago/resource/Trento`

4	italia, autunno, perugiacathedral	`http://mpii.de/yago/resource/Italy`
		`http://mpii.de/yago/resource/wordnet_fall_115236859`
		`http://mpii.de/yago/resource/Perugia_Cathedral`

was derived from a sample set of tags from our corpus of Flickr images from the Trentino region, and can be used to categorise 78% of all tags in the corpus. This categorisation allows us to search for entities that have a geographical location because we can filter out entities of type date, weather and activity which is not specific to one geographical location.

In order to categorise the identified matches we found in YAGO2 (from the previous step), we look up the identified entities in DBPedia and Geonames. This is possible because YAGO2 incorporates the property "hasWikipediaUrl" which DBPedia and Geonames both reference. In order to identify the matches' categories we recurse through the type hierarchies of DBPedia and compare lexically the hierarchies. We also map the Geonames *feature code* which categorises towns, regions, countries, mountains and other landscape features, with our categories. We add any entities that we cannot categorise to the 'unknown' set.

In our Trento Cathedral example, we categorise the entities identified from YAGO2 (see the YAGO2 Matches in Table 1) and the tags from similar images (see the tag cloud shown in Figure 1). Table 2 shows the selected categories and corresponding properties that were used to infer these categories.

3.4 Geographic Distance Constraints

Using the categories assigned to the tags in step 3, we aim to verify whether the entities (identified in step 2) which have a geographical location, are located within a certain distance from the predicted location (from step 1). We predefine acceptable maximum distances for entities of type city, region and mountain and found that through preliminary testing that these were suitable values (see Table 3). It is possible to evaluate the height of buildings using the "heightStories" and "floorCount" properties in DBPedia. We approximate a viewing distance for buildings using these properties. Based on an empirical evaluation, with every

Table 2. Entities, their hierarchies and category.

Entity or Tag	Hierarchy	Category
Trento Cathedral	Cathedrals in Italy, Cathedral, Church, Place of Worship, Building	Building
Trento	Cities and towns in Trentino-Alto Adige, City	Town/City
Trentino	Provinces of Italy, State, District, Region	Region
Italy	Italian-speaking countries, Country	Country
Luglio (July)	Months, Calendar Month, Time Period	Date
Autunno (Autumn)	Season	Season
Sunny	Weather	Weather
Piazza	Public Square, Tract, Location	Place
perugiacathedral	Cathedrals In Italy, Cathedral, Church, Place of Worship, Building	Building
NikonD50	Nikon DSLR Camera, Camera, Photographic Equipment, Equipment, Artifact, Object, Physical Entity, Entity	Unknown
cattedraledisanvigilio	Cathedrals in Italy, Cathedral, Church, Place of Worship, Building	Building
katedralodesantkavigilio	Cathedrals in Italy, Cathedral, Church, Place of Worship, Building	Building

floor we estimate that it is possible to see a building from a further 5 meters away. This is, however, an approximation because it will differ with landscape features such as elevation, the average height of buildings around the building in the query image, and the height of the floors.

Our approach cannot guarantee that recommended entities are contained within the query image because an entity might be within range of the estimated location but it may not be within sight of the camera because other objects may block the view or recommended entities might be located in a different direction. However, we make our recommendation because images matched with step 1 contain reference to these entities. Therefore, we hypothesise that there is a high likelihood that these recommended entities are depicted in the query image.

Table 3. Maximum allowed distances.

Category	Maximum Distance (KM)
place	0.5
city	7
region	30
mountain	50

In our Trento Cathedral example, the entity Duomo of category building has 5 floors and is 10 meters from the estimated geolocation. Using our approach we validate that the Duomo is within our specified range. In the tags related to the similar images, we identify that the building “Perugia Cathedral” has 6 floors and is 343.5 kilometers away from the estimated location. Therefore, we do not recommend URI of this building because its is not within range.

3.5 Recommendation Expansion

In the final step of our approach, we aim to derive further matches by expanding our search terms. Specifically, we expand all non place entities (excluding entities of the type city, town, region and country) with the the closest place name, using the pattern `[place name][non place entity]`. This allows us to disambiguate entities that are common to many places, such as town halls, police stations and libraries. We then check whether the matches are located close to the estimated coordinates. In our Trento Cathedral example, the tag piazza is expanded to “Trento piazza” which is linked to by YAGO2 by the “isCalled” property to the entity “Trento-Piazza_del_Duomo”. We then validate that the “Trento-Piazza_del_Duomo” is categorised with place and is within the geographic distance range of 0.5km from the estimated geographical location. In Table 4 we detail the extended tags which we attempt to match to YAGO2. Table 5 details the recommended URIs for our example.

Table 4. Extended Tags and URIs.

Extended tag [place][non place entity]	URI
Trento Cathedral	`http://www.mpii.de/yago/resource/Trento_Cathedral`
Trento Piazza	`http://www.mpii.de/yago/resource/Trento_Piazza`

Table 5. Recommended Entities and URIs.

Recommended Entity	URI
Province of Trento	`http://en.wikipedia.org/wiki/Province_of_Trento`
Trento	`http://en.wikipedia.org/wiki/Trento`
Italy	`http://en.wikipedia.org/wiki/Italy`
Trento Cathedral	`http://www.mpii.de/yago/resource/Trento_Cathedral`
Trento Piazza	`http://www.mpii.de/yago/resource/Trento_Piazza`

4 Examples and Discussion

In this section, we discuss the recommended URIs for four example query images. The dataset of partially annotated images used as the basis to testing the approach was crawled from Flickr. We first downloaded approximately 150,000 geo-tagged images that were within the bounds of the province of Trentino, an area of 2,397 square miles in the north of Italy. This set was then enriched with an additional 400,000 images with keyword tags relating to the Trentino region.

In total our image set consists of 472,565 images because of the intersection of these images sets[5]. We randomly sampled 3,250 images from this set and manually identified the theme of the image (see Table 6).

Table 6. Sample of topics from 3,250 images

Theme	Percentage (%)
People	31.2
Landscape	26.2
Houses	16.5
Animals	12.5
Churches	5.8
Other	5.6
Transport	1.6
Trento Cathedral	0.6
Buonconsiglio Castle	0

We considered using standard image sets such as the European Cities 50K[22] dataset and MIRFlickr[19]. However, we wanted to include images from the surrounding area because often they are the most similar visually, areas typically have a particular style due to tradition and age of the area. The European cities set contains images from different cities and does not include whole regions. Similarly we chose not to use the MIRFlickr image set as it was not suitable because our approach requires both tags to identify landmarks and geotags to disambiguate between landmarks, and 88.948% of the images in MIRFlickr did not contain geotags. Whereas our image set contains over double the number of geotagged images at 23% compared to 11%. To the best of our knowledge there were no suitable datasets which contained a complete area of geotagged images, or that contained a ground truth of URIs associated with each image.

4.1 Example 1

The first query image depicts Trento Cathedral, the Neptune Fountain and a plaza, and the visual feature search returned the images that depict the cathedral (see Figure 2). The seventh similar image is a photograph of Trento Cathedral and that image is not geographically close to the landmark's actual location. While estimating the geographical location of the image, the seventh image's geotags are removed by the threshold described in Section 3.1, and therefore does not effect the recommended URIs. Our approach correctly identifies that the query image contains Trento Cathedral and recommends the URIs for the Italian and English wikipedia URIs for the cathedral because the tag cloud contains 'trentocathedral', 'trento' and 'cathedral'. It also recommends URIs relating to the region, such as town, region, and country, and URIs relating

[5] Our image set can be downloaded here: `http://degas.ecs.soton.ac.uk/~hp07r/fullcollection.csv`

to ecclesiastical buildings, notably it recommended URIs about the cathedral's Pope (see following list).

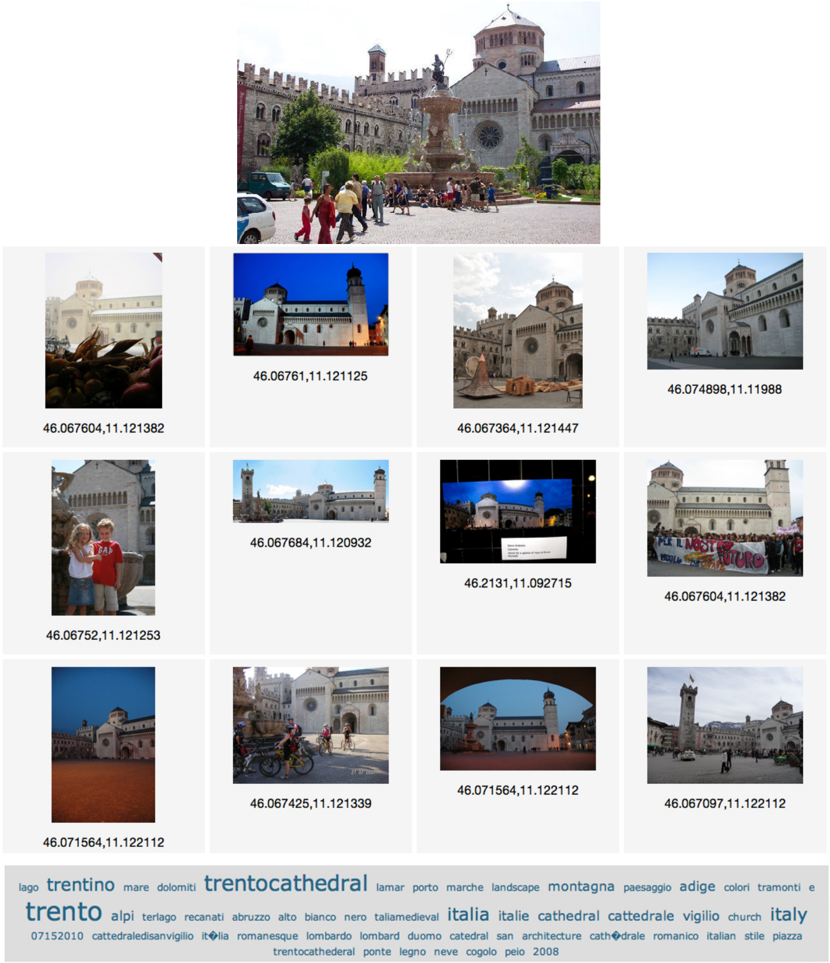

46.067604,11.121382
46.06761,11.121125
46.067364,11.121447
46.074898,11.11988
46.06752,11.121253
46.067684,11.120932
46.2131,11.092715
46.067604,11.121382
46.071564,11.122112
46.067425,11.121339
46.071564,11.122112
46.067097,11.122112
lago trentino mare dolomiti trentocathedral lamar porto marche landscape montagna paesaggio adige colori tramonti e
trento alpi terlago recanati abruzzo alto bianco nero taliamedieval italia italie cathedral cattedrale vigilio church italy
07152010 cattedraledisanvigilio it�lia romanesque lombardo lombard duomo catedral san architecture cath�drale romanico italian stile piazza
trentocathederal ponte legno neve cogolo peio 2008

Fig. 2. The query image of Trento Cathedral, similar images and tag cloud.

1. `http://en.wikipedia.org/wiki/Trento_Cathedral`
2. `http://it.wikipedia.org/wiki/Cattedrale_di_San_Vigilio`
3. `http://it.wikipedia.org/wiki/Provincia_autonoma_di_Trento`
4. `http://www.mpii.de/yago/resource/Italy`
5. `http://en.wikipedia.org/wiki/Italy`
6. `http://en.wikipedia.org/wiki/Trento`
7. `http://it.wikipedia.org/wiki/Church`
8. `http://en.wikipedia.org/wiki/Alps`

9. `http://en.wikipedia.org/wiki/Cathedral`
10. `http://it.wikipedia.org/wiki/Trento`
11. `http://it.wikipedia.org/wiki/Cattedrale`
12. `http://it.wikipedia.org/wiki/The_Church`
13. `http://en.wikipedia.org/wiki/Province_of_Trento`
14. `http://en.wikipedia.org/wiki/Church`
15. `http://www.mpii.de/yago/resource/Province_of_Trento`
16. `http://en.wikipedia.org/wiki/Pope_Vigilius`
17. `http://it.wikipedia.org/wiki/Papa_Vigilio`
18. `http://it.wikipedia.org/wiki/Cathedral`
19. `http://en.wikipedia.org/wiki/Trentino`
20. `http://www.mpii.de/yago/resource/Trento`
21. `http://en.wikipedia.org/wiki/Mountain`
22. `http://en.wikipedia.org/wiki/Alto`

4.2 Example 2

The second query image depicts Trento Cathedral, and the visual feature search correctly matched seven images of the cathedral (see Figure 3). From the tag cloud we can see that one or more of the similar images has been incorrectly tagged with ‘Buonconsiglio’ and ‘Castle’. These tags refer to Buonconsiglio Castle which is approximately 700 meters from Trento Cathedral. In step four of our approach, we disambiguate between places of interest when there is a distance greater then 0.5km. However, in this case, our approach was unable to disambiguate between the two places of interest because all the geotagged images were within 0.5km of Trento Cathedral (as defined on Geonames) and contained tags relating to both the cathedral and castle. If the image tagged with ‘Buonconsiglio’ was geographically located at the castle, then our approach would have only recommended URIs relating to the cathedral. Our approach recommended the URIs in example 1 and those in the following list, and recommended URIs that relate to both Buonconsiglio Castle and Trento Cathedral.

1. `http://it.wikipedia.org/wiki/Castello_del_Buonconsiglio`
2. `http://en.wikipedia.org/wiki/Castello_del_Buonconsiglio`
3. `http://it.wikipedia.org/wiki/Castello`
4. `http://www.mpii.de/yago/resource/Trento_Cathedral`

4.3 Example 3

The third query image also depicts Trento Cathedral (see Figure 4). The visual feature search matched three images but only one of the images depicted Trento Cathedral. Non of these images were tagged, therefore our approach could not find or expand any tags to look up entities in YAGO2, DBPedia or Geonames.

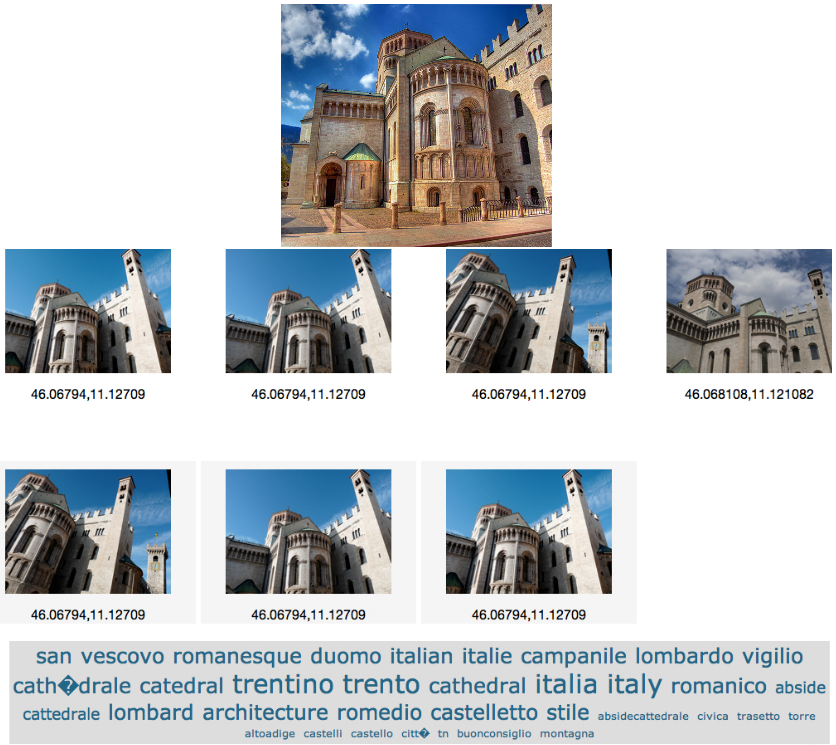

46.06794,11.12709
46.06794,11.12709
46.06794,11.12709
46.068108,11.121082
46.06794,11.12709
46.06794,11.12709
46.06794,11.12709
san vescovo romanesque duomo italian italie campanile lombardo vigilio
cath�drale catedral trentino trento cathedral italia italy romanico abside
cattedrale lombard architecture romedio castelletto stile absidecattedrale civica trasetto torre
altoadige castelli castello citt� tn buonconsiglio montagna

Fig. 3. The query image of Trento Cathedral, similar images and tag cloud.

4.4 Example 4

The fourth query image depicts Buonconsiglio Castle. The visual feature search returned over 20 images. Figure 5 shows the first eight images which are the most similar to the query image. The first eight images all depict the castle. The visual feature search also returned images of Trento Cathedral, hence the tag cloud contains tags about the cathedral: cathedral, catedral, cathdrale, cattedrale, and vigilio. Unlike our second example, our approach was able to disambiguate between the castle and cathedral because the similar images were correctly geotagged within 0.5km from the photographed landmark. Our approach expanded the tag Buonconsiglio with castle (see Section 3.5), because it determined that castle was a type of building, and thus was able to identify the wikipedia URI `http://en.wikipedia.org/wiki/Buonconsiglio_Castle`. The following list contains our approach's recommended URIs.

1. `http://en.wikipedia.org/wiki/Buonconsiglio_Castle`
2. `http://it.wikipedia.org/wiki/Castello_del_Buonconsiglio`
3. `http://en.wikipedia.org/wiki/Castello_del_Buonconsiglio`
4. `http://it.wikipedia.org/wiki/Provincia_autonoma_di_Trento`
5. `http://en.wikipedia.org/wiki/Trento`

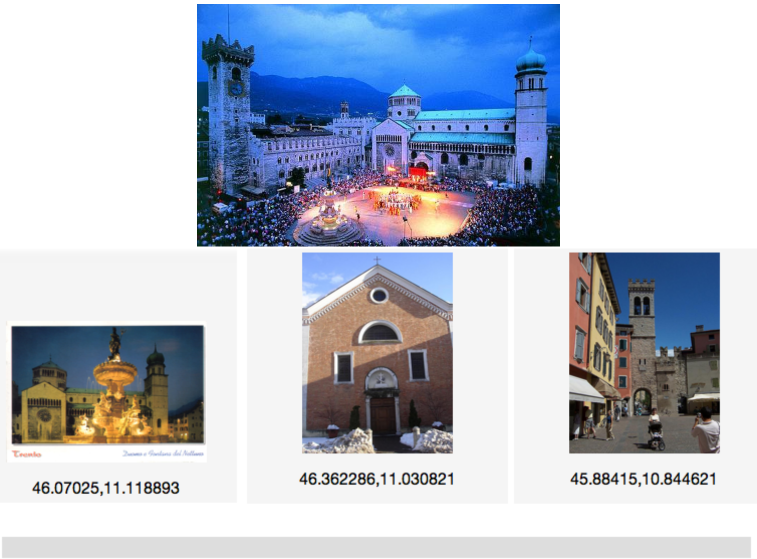


Fig. 4. The query image of Trento Cathedral, similar images and tag cloud.

Our approach can be hindered by the quality of information in the semantic knowledge stores — the set of tags and the coordinates. The examples discussed in this section show that without approximately correct coordinates or tags, our algorithm will not be able to identify and recommend accurate semantic tags. Nor will our approach be able to validate the coordinates of places of interest, if the knowledge base does not contain them.

5 Conclusion

In this paper, we present an algorithm to recommend URIs that represent the visual content of an image and focus on identifying places of interest using geographical information from YAGO2, DBPedia, and Geonames. In order to use these knowledge sources, we use large-scale image matching techniques to find similar images that are then used to estimate geo-coordinates and potential tags.

The four examples show that the quality our results highly depends on the quality of the image matching techniques and the reference corpus. Our approach

[6] `http://www.mpii.de/yago/resource/Italy`
[7] `http://en.wikipedia.org/wiki/Italy`
[8] `http://it.wikipedia.org/wiki/Castello`
[9] `http://it.wikipedia.org/wiki/Trento`
[10] `http://www.mpii.de/yago/resource/Trento`
[11] `http://en.wikipedia.org/wiki/Trentino`
[12] `http://en.wikipedia.org/wiki/Province_of_Trento`
[13] `http://www.mpii.de/yago/resource/Province_of_Trento`

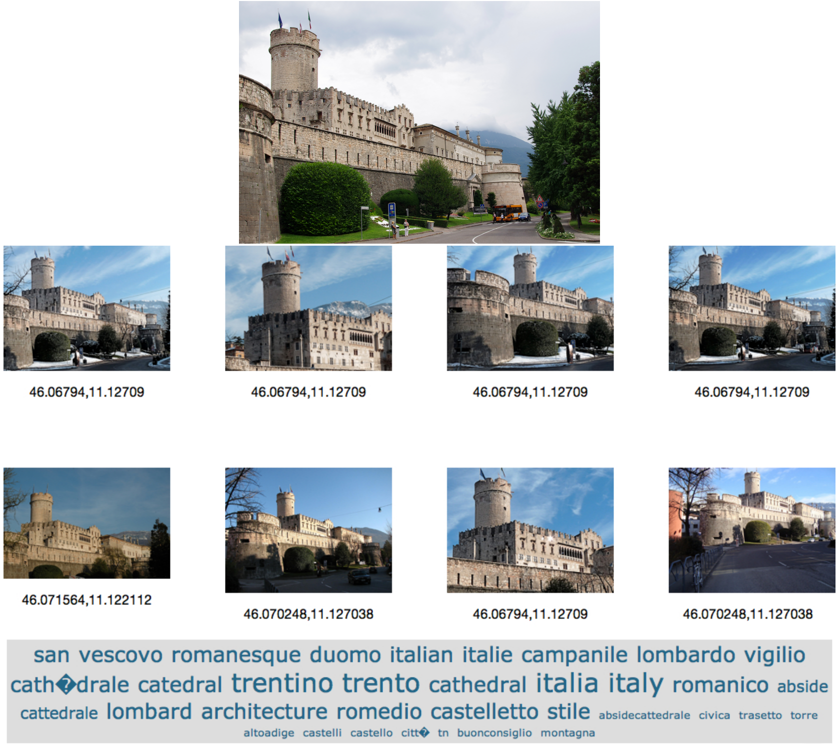

46.06794,11.12709
46.06794,11.12709
46.06794,11.12709
46.06794,11.12709
46.071564,11.122112
46.070248,11.127038
46.06794,11.12709
46.070248,11.127038
san vescovo romanesque duomo italian italie campanile lombardo vigilio
cath�drale catedral trentino trento cathedral italia italy romanico abside
cattedrale lombard architecture romedio castelletto stile absidecattedrale civica trasetto torre
altoadige castelli castello citt� tn buonconsiglio montagna

Fig. 5. The query image of Buonconsiglio Castle, similar images and tag cloud.

performs best when there are accurate tags and geotags and this is not always the case with collections of images. For future work, we plan to develop approaches that consider how to better handle keyword tags and geotags that are incorrect.

Acknowledgments

This work was funded by the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreements n° 270239 (ARCOMEM), 287863 (TrendMiner) and 231126 (LivingKnowledge) together with the LiveMemories project, graciously funded by the Autonomous Province of Trento (Italy).

References

1. J. Hoffart, F. M. Suchanek, K. Berberich, E. Lewis-Kelham, G. de Melo, and G. Weikum, "YAGO2: Exploring and Querying World Knowledge in Time, Space, Context, and Many Languages," in *Proceedings of the 20th International Conference Companion on World Wide Web*. ACM, 2011, pp. 229–232.
2. S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives, "Dbpedia: A Nucleus for a Web of Open Data," *The Semantic Web*, pp. 722–735, 2007.

3. B. Vatant and M. Wick, "Geonames Ontology," *GeoNames, Accessed*, vol. 6, 2009.
4. H. Packer, S. Samangooei, J. Hare, N. Gibbins, and P. Lewis, "Event Detection using Twitter and Structured Semantic Query Expansion," in *CIKM2012 - The First International Workshop on Multimodal Crowd Sensing*, 2012.
5. H. Packer, A. Smith, and P. Lewis, "MemoryBook: Generating Narrative from Lifelogs," in *Hypertext2012 - The Second International Workshop on Narrative and Hypertext Systems*, 2012.
6. E. Moxley, J. Kleban, and B. Manjunath, "Spirittagger: a Geo-Aware Tag Suggestion Tool Mined from Flickr," in *Proceeding of the 1st ACM international Conference on Multimedia Information Retrieval*, 2008, pp. 24–30.
7. D. G. Lowe, "Object Recognition from Local Scale-Invariant Features," *IEEE International Conference on Computer Vision*, vol. 2, p. 1150, 1999.
8. E. Moxley, J. Kleban, J. Xu, and B. Manjunath, "Not all Tags are Created Equal: Learning Flickr Tag Semantics for Global Annotation," in *IEEE International Conference on Multimedia and Expo.*, 2009, pp. 1452–1455.
9. L. Kennedy and M. Naaman, "Generating Diverse and Representative Image Search Results for Landmarks," in *Proceeding of the 17th International Conference on World Wide Web*, 2008, pp. 297–306.
10. M. Maala, A. Delteil, and A. Azough, "A conversion process from flickr tags to rdf descriptions," in *BIS 2007 Workshops*, 2008, p. 53.
11. H. Kawakubo and K. Yanai, "GeoVisualRank: a Ranking Method of Geotagged Images Considering Visual Similarity and Geo-Location Proximity," in *Proceedings of the 20th International Conference Companion on World Wide Web*, 2011.
12. B. Sigurbjörnsson and R. van Zwol, "Flickr Tag Recommendation Based on Collective Knowledge," in *Proceeding of the 17th International Conference on World Wide Web*, 2008, pp. 327–336.
13. J. Tang, S. Yan, R. Hong, G.-J. Qi, and T.-S. Chua, "Inferring Semantic Concepts from Community-Contributed Images and Noisy Tags," in *Proceedings of the 17th ACM International Conference on Multimedia*, 2009, pp. 223–232.
14. J. Sivic and A. Zisserman, "Video Google: A Text Retrieval Approach to Object Matching in Videos," in *ICCV*, October 2003, pp. 1470–1477.
15. J. S. Hare, S. Samangooei, and D. P. Dupplaw, "OpenIMAJ and ImageTerrier: Java libraries and tools for scalable multimedia analysis and indexing of images," in *Proceedings of ACM Multimedia 2011*, ser. MM '11. ACM, 2011, pp. 691–694.
16. J. Hare, S. Samangooei, D. Dupplaw, and P. Lewis, "Imageterrier: An extensible platform for scalable high-performance image retrieval." in *The ACM International Conference on Multimedia Retrieval (ICMR 2012)*, 2012.
17. D. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, vol. 60, no. 2, pp. 91–110, January 2004.
18. J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object Retrieval with Large Vocabularies and Fast Spatial Matching," in *CVPR*, 2007.
19. M. J. Huiskes and M. S. Lew, "The MIR flickr retrieval evaluation," in *Proceeding of the 1st ACM international conference on Multimedia information retrieval*, 2008.
20. H. Jegou, M. Douze, and C. Schmid, "Hamming Embedding and Weak Geometric Consistency for Large Scale Image Search," in *Proceedings of the 10th European Conference on Computer Vision*, 2008, pp. 304–317.
21. D. Nistér and H. Stewénius, "Scalable recognition with a vocabulary tree," in *CVPR*, 2006, pp. 2161–2168.
22. Y. Avrithis, G. Tolias, and Y. Kalantidis, "Feature map hashing: Sub-linear indexing of appearance and global geometry," in *in Proceedings of ACM Multimedia*, Firenze, Italy, October 2010.

Ageing Factor: a Potential Altmetric for Observing Events and Attention Spans in Microblogs

Victoria Uren[1] and Aba-Sah Dadzie[2]

[1]Aston Business School, Aston University, The Aston Triangle, Birmingham, UK
[2]Department of Computer Science, 211 Portobello, The University of Sheffield, Sheffield, UK

We present an initial examination of the (alt)metric ageing factor to study posts in Twitter. Ageing factor was used to characterize a sample of tweets, which contained a variety of astronomical terms. It was found that ageing factor can detect topics that both cause people to retweet faster than baseline values, and topics that hold people's attention for longer than baseline values.

1 Introduction

Our long-term goal is to study public communication about science: understanding what the public thinks about scientific research is important for many reasons, from developing science policy, through making the case for technological developments and determining the impact of research, to simply being able to characterize the vibrant public discourse which marks a healthy society, in which science plays as important a part as politics, popular culture or the arts. The new media of the social web, which are open to all, offer a fresh insight into public opinion to supplement the surveys, and so forth, used in Public Understanding of Science (PUS) research [20].

Because of the scale of data available (Twitter claims there are 340M tweets per day[1]), quantitative metrics are needed to aggregate the contributions of many individuals. Informetrics research has developed and used quantitative measures to study scholarly communication in traditional media for decades. Informetric methods have been shown to transfer to communication on the web [1] and latterly social media / Web 2.0 [2], in which field they are coming to be known as altmetrics. Altmetrics adapt tried and tested informetric methods to the analysis of scientific communication in social media. Used along with natural language methods, such as clustering and summarization, we believe they have much to offer analysts. Our aim, in the experiments reported in this paper, was to study a metric called ageing factor as a means to characterize whether people's interest in discussion of scientific topics on Twitter is sustained or transient.

[1] http://blog.twitter.com/2012/03/twitter-turns-six.html

1.1 Science in Social Media

While reservations about the use of social media in formal work still exist [3], many scientific communities have embraced social media as a mode of communication [4]. In one survey [5], Twitter was one of the highest ranked services for semantic web researchers, but other communities favour different services, e.g., of informetrics researchers only 16% had Twitter accounts while 70% used LinkedIn [2]. Ponte & Simon [4] similarly found, in a survey of researchers from different fields, that nearly 18% used micro-blogging services such as Twitter, while 40% used science-based blogs and social networks. The ways in which scientific communities use social media vary. In computer science, delegates tweet or blog running commentary on conferences, opening up the proceedings to delegates in other rooms as well as colleagues who cannot attend [5] [6]. Whereas in biology, Mandavilli reports examples of intensive public critique of contentious articles [7].

However, scientists play only one part in the bigger picture of science communication - science organizations, journalists (both science and public-interest media), lobbyists and the general public also have important contributions to make. The current ethos of science communication, as discussed by Nisbet & Scheufele [8] and many others, advocates engaging the public in dialogue. Social media support open access to discussion, and for science organizations, Web 2.0 has become an essential part of their public relations operation. As one white paper boldly states "*the people formerly known as the audience are now at the center of media*" [9]. This is echoed by Eysenbach [10], who describes the "traces" left by both scientists and the lay public as they interact with information on the web. Individual scientists are increasingly aware of the public engagement agenda: Ponte and Simon [4] report scholars' desire to make greater use of Web 2.0 methods for peer review and communication of research. Letierce [5] has shown that although researchers' main motivation for using Twitter is to communicate with members of their own community (89%), some are also trying to reach general audiences (45.9%). Consequently, we argue that it is necessary to expand the scope of analysis of science communication in social media beyond the boundaries of scientists' communication with each other to encompass wider public communication about science.

Compared to the numerous works on politics and marketing, relatively few studies exist about the public's (as opposed to researchers') scientific communication in microblogs. Hubmann-Haidvogel et al. [11] present a visualization tool using climate change posts as a use case, and Chew and Eysenbach [12], report the timeliness of social media for highlighting trends in the development of pandemics. In our own experience [13], the lack of research may be because scientific topics typically have few posts compared to current affairs, popular culture etc., so that simple trend spotting methods can be ineffective. Furthermore, there is a high level of noise, with many "scientific" search terms being used in non-scientific contexts [13]. Chew and Eysenbach [12], and Weller and Puschmann [14] also recognise the negative influence of noise in Twitter data. Chew and Eysenbach remark that "*spam and popular news articles that contained key phrases can influence search results and create peaks in activity that may not be reflective of the concept*"[12]. Both suggest the use of advanced natural language processing to support the identification of tweets containing scientific information and to filter out non-relevant or spurious use of domain-specific terms.

Weller et al. [15] have identified three classes of scientific microposts for their study of communication in scientific conferences: posts (tweets on Twitter) with links to scientific content, posts published by a scientist, and posts with a science related hashtag. To these we add posts that contain scientific terms used within a scientific context, as a more general definition of the sorts of posts we need to identify in order to analyse general public communication about science.

1.2 Ageing Factor

Our earlier work [13] which looked at changes in the usage of a sample of scientific terms over time, determined that the basic trend spotting approach, which looks for peaks of tweets occurring for a term on a timeline, is not effective for scientific data because the level of tweeting about science is very low compared to the level of tweets on popular culture. Furthermore, scientific terms are frequently used in non-scientific contexts. The combination of these two factors means that it is difficult to reliably identify peaks of discussion about science topics, because they are small and obscured by noise (irrelevant tweets using the same words). In order to study public communication about science in public media, we need an altmetric which is sensitive even on moderate sized samples of data, because big trend peaks, such as those seen in pandemics, will be relatively rare. We are interested in studying microblogging, for which time is a critical parameter, therefore, a metric which looks at temporal effects is needed.

The metric we test in the experiments presented here, called aging factor, is based on a well-established informetric measure. We follow the convention in which, by analogy, retweets are treated as citations (tweetations) by researchers working on altmetrics. For example, Eysenbach calculated a range of metrics for retweets of announcements by the Journal of Medical Internet Research about the publication of new papers [10]. He uses these to determine whether initial interest indicated by retweets can be correlated with later citation rates. The metrics Eysenbach used include tweet half-life defined as "*the point in time after publication by which half of all tweetations of that article within the first n days occur*".

Half-life is similar to the ageing factor metric used in this paper with a number of important differences, which we consider make ageing factor more suitable for the kind of data we are studying. Half-life takes what is called the *diachronous view*, i.e. the metric observes a fixed set of documents, such as one year's articles in a given journal or one year's tweets from a particular organization). It is therefore useful for organizations which want to judge the impact of their own tweets and are monitoring the occurrence of their Twitter name in retweets on an on-going basis. This is necessary in order to harvest every tweetation of a particular tweet in the first *n* days. By contrast, we want to take a snapshot of general discussion on scientific topics in a given time period. This is what is termed the *synchronous view* and requires a metric which does not rely on the originating tweet being present in the sample. For this we adapted Avremescu's ageing factor measure as presented in [16], changing the counts of citations to retweets and changing time windows measured in years to windows measured in hours to suit the fast pace of communication on Twitter. Aging factor, *AF,* is defined below, where *i* is the cut-off time in hours, *k* is the number of retweets originating at least *i* hours ago and *l* is the number of retweets originating less than *i* hours ago.

$$AF = \sqrt[i]{\frac{k}{k+l}}$$

We examined two values of *i*, *i*=1 giving the one hour ageing factor (*1hAF*), and *i*=24 giving the 24 hour ageing factor (*24hAF*). A convenient feature of 1hAF is that it is simply the ratio of retweets in a sample that originated more than one hour after the original *createdAt* time over the total number of retweets in the sample; this makes it easy to understand. The 24hAF is the 24th root of the similar ratio for a 24 hour cut-off. In either case, *AF* values are produced in the range 0-1 with higher values indicating more retweets originating after the cut-off.

2 Experiments

In the context of communication on Twitter, low values of AF would suggest a flurry of activity typical of a trending topic, such as might happen following the posting of tweets about an exciting topic. This might be a special event - in our experiments we looked at retweeting about meteor showers. This fits with the findings in [10] that even for the most interesting or highly cited articles, twitter citations quickly fall off soon after announcement of publication. On the other hand, we interpret high values of AF as an indication that people have shown sustained interest in a topic and continue to read and retweet posts for a long (in terms of Twitter) time after they appear. We argue that, for science, being able to show the public has a long attention span for ongoing developments in a field is as important as showing you can get a reaction to a hot news item. Based on these interpretations of what ageing factor could tell us, we made two assumptions about how to interpret ageing factor.

Assumption 1: ageing factors for topics which concern special events will be lower than suitable baselines.

Assumption 2: ageing factors which are higher than suitable baselines are associated with topics in which interest is sustained over time.

The question of what constitutes a "suitable" baseline therefore arises. Unfortunately no benchmark corpus of Twitter presently exists (plans for a corpus to be held at the United States Library of Congress are believed to be underway at the time of writing). In this experiment, we have taken a pragmatic approach. We know there is a high level of noise in the samples (see table 1) - the majority of posts for selected terms are *not* about science. Therefore, we take the superset of tweets as a sample of general usage of that term on Twitter at that time, and use ageing factors for these sets as our baselines.

We chose the topic of astronomy for the experiment, because it has an enthusiastic following of amateur stargazers who own their own telescopes and are interested in observing events in the night sky for themselves. This led us to believe that it would be possible to harvest posts from Twitter, which discussed astronomical events and might provide evidence about the validity of assumption 1. We collected data on two nights when meteor showers were expected. Our data collection harvested tweets containing a broad range of astronomical terms in order to compare ageing factors for meteor showers with those for other astronomical topics.

2.1 Experiment 1 – Geminid Meteor Shower

As in our previous experiment [13], the UNESCO Thesaurus[2] was used as a source of scientific terminology. The starting point was the terms under the subheading "*Astronomical Systems*". We used 32 of 33 terms for the initial filtering, a mix of single words and two word phrases (see Table 1). The 33rd term, *Time*, produced an unacceptably high level of noise and was therefore removed from the set after an initial test run of the harvesting program. Using the public Twitter stream, two data sets were collected: a training set, comprising 8980 tweets collected between Dec 14th 2011 at 22:36 GMT and Dec 14th 2011 at 23:18 GMT, and a test set, comprising 81891 tweets collected between Dec 14th 2011 at 23:18 GMT and Dec 15th 2011 at 03:30. Dec 13-14th 2011 were the nights on which the annual Geminid meteor shower was expected to take place.

Table 1. Occurrence of retweets (RT) containing UNESCO terms in the training data, and the number of retweets judged to have scientific content (Sci) (where RT > 98).

UNESCO Term	RT	Sci	UNESCO Term	RT	Sci
Celestial bodies	0		Solar activity	0	
Cosmic matter	0		Solar disturbances	0	
Interstellar matter	0		Sunspots	0	
Galaxies	1		**Stars**	**174**	**7**
Stellar systems	0		Quasars	0	
Interstellar space	0		**Universe**	**99**	**5**
Black holes	2		Cosmos	5	
Meteorites	0		Astronomy	14	
Comets	9		Astrophysics	0	
Meteors	13		Gravitation	1	
Solar system	1		Celestial mechanics	0	
Planets	28		Cosmology	0	
Earth	**213**	**8**	Cosmogeny	0	
Satellites	1		**Space**	**166**	**27**
Moon	**241**	**9**	Outer space	3	
Sun	**565**	**6**	Space sciences	0	

[2] http://www2.ulcc.ac.uk/unesco/thesaurus.htm

For the AF calculations we needed to pick terms with reasonable levels of retweets. Our previous experiment [13] with a range of scientific terms, lead us to predict that many of the tweets that used UNESCO terms would not have scientific content. This proved to be true for the astronomical terms (see Table 1). Of the UNESCO terms identified in retweets in significant numbers, most are words used in daily life, which do not necessarily have an astronomical meaning: Sun, Moon, Stars etc. are used in a multitude of colloquial ways. The proportion of retweets judged to be scientific, from the six terms categorised, was 0.043. This is substantially lower than levels reported elsewhere: e.g., Mejova and Srinivasan [17] report 0.389 tweets judged to be topical, for a collection of tweets with the categories movies, music albums, smart phones, computer games, and restaurants and note that this is low compared to 0.60 for their sample of blogs. They identify "*the need for more precise retrieval strategies for Twitter*". We suggest this is even more important for scientific communication.

We considered terms with 99 or more retweets in the test data to be worth considering in the experiment. These were classified by reading the tweets and making a judgement about whether or not they had scientific content. For example, "*when I was little I thought the sun and moon followed me around everywhere!"* was judged not scientific, whereas "*If the Sun exploded we wouldn't know for 8m 20s. Light & gravity take that long to reach us. Then we'd vaporize"* was judged scientific. Some retweets needed more research, for example, "*RT @VirtualAstro: Make sure You watch a Night with the stars with your illustrious leader on Sunday night :)"* was judged scientific after establishing the @VirtualAstro describes himself as "*The Basil Fawlty of Astronomy, Science, Nature and more.*"; he fits the profile of an amateur stargazer. As can be seen from Table 1, this classification exercise made clear the low level of scientific retweets.

Having identified UNESCO terms for which we could harvest reasonable levels of scientific retweets (albeit along with significant amounts of noise), we selected two baselines: the UNESCO thesaurus term *Space*, and a compound term we labelled Astro, which bundled together the UNESCO terms Earth, Moon, Sun, Stars, Universe and Space. In their raw condition, these both contain high levels of non-scientific usage of terms. Therefore, each can be considered as a sample of general use of those terms on Twitter at the sampled point in time. Subsets of the baseline, selected to filter out noise and represent scientific usage of the terms then had to be extracted.

Identifying Scientific Retweets.

Ideally, we would use a natural language processing method to identify scientific use of the terms. However, these experiments have the objective of testing whether AF is an appropriate metric for studying scientific communication. Therefore, we took a simple approach to identifying scientific retweets using SQL queries to reduce the noise in samples by adding narrower terms. We accept that this approach, based on human interpretation of the language of the domain, has limitations for practical implementation on the large scale and will need to be replaced in future work with advanced NLP methods as advocated by [12].

The training data was a small enough sample to be analysed by hand. The retweets that had previously been identified as scientific were reviewed and topically related terms, which co-occurred with more than one UNESCO term, were identified. The terms were then sorted into topical queries. For example, terms related to space exploration equipment (e.g. Hubble) were in one set and terms which would be ambiguous (e.g. program) were all grouped together in another. One search (e.g. Space and Bodies+) added the names of the planets (plus Pluto) based on background knowledge.

Table 2 presents short form versions of the queries used, in which | represents OR, and standard parts of the query have been omitted for clarity. An actual search statement for the search *Space AND sci* for the training data set would read:

```
SELECT statusid, createdat, retweetid, retweetcreatedat
FROM 'twitter' WHERE (retweetid != "" AND batch =
"1323902158593" AND (text like '%space%') AND (text like
'%nasa%' OR text like '%science%' OR text like
'%station%' OR text like '%soyuz%' OR text like
'%satellite%' OR text like '%hubble%' OR text like
'%interstellar% 'OR text like '%program% 'OR text like
'%physics% 'OR text like '%plane% 'OR text like
'%voyager%')) ORDER BY retweetid ASC;
```

Table 2. Terms used in searches

Search label	**Terms**
Batch	Batch number only
Space	space
Space AND sci	Space AND (nasa|science|station|soyuz|satellite|hubble|interstellar|program|physics|plane|voyager)
Space AND gear	Space AND (nasa|soyuz|satellite|spaceflight|orbit|hubble|telescope|spacecraft|voyager)
Space AND amb	Space AND (agency|program|plane|rock|beam|aircraft|station|aero|astro|launch|deep|outer|travel)
Space AND bodies	Space AND (interstellar|black hole|comet|moon|geminid)
Space AND bodies+	Space AND Bodies AND (planet|mercury|venus|mars|jupiter|saturn|neptune|uranus|pluto)
Astro	Earth|moon|sun|stars|universe|space
Astro AND events	Astro AND (meteor|shooting star|launch|phaethon|geminid)
Astro AND @	Astro AND (@universetoday|@sciencemagazine|@brainpicker|@NASA_GoddardPix|@doctorjeff|@earthskyscience|@anditurner|@Sky_Safari, @VirtualAstro|@NASAAmes|@NASA_Lunar,)
Astro AND tech	Astro AND (light year|astronomy|galactic|gravity|astronaut)
Astro NOT meteor	Astro AND (nasa|science|astro|hubble) NOT (meteor|geminid|(shooting AND star)
Meteor	meteor|geminid|(shooting AND star)

Table 3 shows both 1hAF and 24hAF for the searches. The 24hAF values for this dataset were all in the range 0.8-0.95 (zero values were assigned when all retweets collected were within the 24 hour window), whereas 1hAF ranged from 0.25-0.65. In general, 24hAF tracks 1hAF. The culture of Twitter places high value on currency, and 24h is a long time for many Twitter users. 24hAF appears to be an insensitive metric and we used 1hAf only for the remainder of the experiments.

1hAF for the training data searches are often based on small samples of retweets, such that just two or three retweets can make a big difference to the 1hAF. For example, the 1hAf value for Space and gear of 0.73 in the training data is based on 11 retweets, and consequently the 0.4 difference compared to 1hAF in the test data (0.33) is unlikely to be significant. Therefore, the following observations look only at test data, and use Batch, Space and Astro test values as baselines. In this first experiment, we made a naïve interpretation of the results, simply looking for values of 1hAF that appeared high or low, then trying to explain them in terms of the content of retweets.

Table 3. Number of retweets, 1hAF and 24hAF values for experiment 1. Baselines for *Space Astro* and the whole batch are **0.34**, **0.37** and **0.37** respectively

Search Label	Training Data			Test Data		
	RT	1hAF	24hAF	RT	1hAF	24hAF
Batch	1526	0.31	0.88	15583	0.37	0.91
Space	166	0.43	0.90	2594	0.34	0.90
Space AND sci	18	0.44	0.89	396	0.33	0.85
Space AND gear	11	0.73	0.90	375	0.33	0.83
Space AND amb	25	0.48	0.87	487	0.37	0.87
Space AND bodies	7	0.57	0	42	**0.45**	0.88
Space AND bodies+	7	0.57	0	315	**0.26**	0.82
Astro	1425	0.31	0.88	14634	0.37	0.91
Astro AND events	25	0.44	0.93	597	**0.26**	0.86
Astro AND @	14	0.43	0.90	23	**0.65**	0
Astro AND tech	16	0.25	0	122	**0.58**	0.90
Astro NOT meteor	29	0.55	0.91	511	0.36	0.88
Meteor	27	0.15	0	364	**0.22**	0.84

For the Space set, two searches have 1hAF values that look different to the baselines: 1hAF for Space AND bodies is increased (0.45 compared to 0.34 in the Space baseline and 0.37 for the batch), 1hAF for Space AND bodies+ is decreased (0.26 compared to the same baselines). For the Astro set of queries, all queries except Astro NOT Meteor show differences when compared to the baseline Astro. Searches with increased 1hAF are: Astro AND @ (0.65 compared to 0.37 for the Astro baseline, but with only 23 retweets in the sample we should be cautious about its significance), and Astro AND tech (0.58). Astro AND events shows decreased 1hAF (0.26). The search Meteor was run to isolate tweets concerning the Geminid meteor shower. As can be clearly seen, it has a low value of 1hAF (0.22 compared to 0.37 for the batch).

Assumption 1 would associate low 1hAF with an event of some kind. The text of the retweets was examined and we found a high level of retweets of "*NASA launch new rover to Mars"* tweet in both Space AND bodies+ (which contains the term Mars) and Astro AND events (which contains the term launch). It seems the high level of retweeting of this post brings the 1hAF down for these two subsets. These initial results were sufficiently encouraging to make us want to study 1hAF in more detail with a larger dataset.

2.2 Experiment 2: Quadrantid Meteor Shower

A larger sample of the public Twitter stream was then collected. This was filtered using the same 32 UNESCO astronomy terms and covers the full 24 hours of the 3rd of January 2012. This was the night on which the annual Quadrantid meteor shower was expected and our aim was to see if 1hAF values were low for this event, as per assumption 1, and whether the time of day matters (it must be dark to see meteors).

Initially we filtered out subsets using the searches we had developed using the training data for the first experiment (see Table 2). The day was divided into four periods 0:00-5:59 GMT (labelled 6), 6:00-11:59 GMT (12), 12:00-17:59 GMT (18) and 18:00-23:59 (24). Figure 1 shows the 1hAF values for these searches.

The first observation is that although the batch baseline 1hAF is steady in the range 0.32-0.37 through the day, the other two baselines each have one quarter of the day when they are high or low (24 for Space and 12 for Astro).

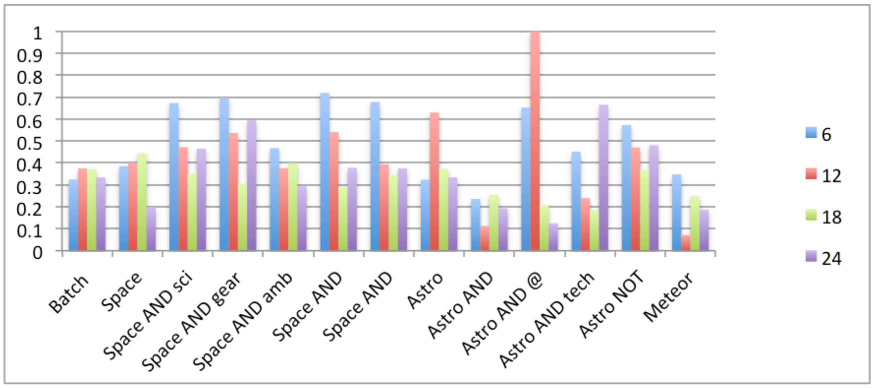


Fig. 1. 1hAF for 4*6 hour periods over the course of 3rd Jan. 2012

As in the previous experiment, the 1hAF values for Astro are more variable than those for Space. For example, both the Astro AND Events 1hAF and the Meteor 1hAF (both of which contain the term *shooting star*) are relatively low compared to the Astro baselines, especially for period 12 (2nd quarter of the day). It would be easy, but incorrect, to infer that 1hAF had identified a flurry of retweets about the meteor shower. Examination of the text of posts in the second quarter show that of 275 total retweets 18 contain the term *quadrantid* while 213 contain the term *wish*.

There are various original tweets, but "*@iQuoteFresh: #IfYouWereMine I'd stop wishing on 11:11's, birthday candles, dandelions and shooting stars...Because I'd have my wish ...*" is a typical example. Noise from non-scientific posts clearly remains an important issue, a fact underlined by examination of the high 1hAF values observed for four of the Space searches. These turned out to be due largely to retweets of variants of humorous posts on the lines of "*Oh really? You need space? You might as well join NASA.*", for which the original tweets were more than one hour old.

The searches were based on the sample of training data collected about two weeks earlier around the Geminid meteor shower. It seems that even in this short time, the ways terms were being used had changed. We therefore took further steps to remove noise from our samples. Three astronomical events that took place around the 3rd of Jan. 2012 were used as background knowledge to add narrower terms to three of the original searches. The events were the Quadrantid meteor shower on the night of 3-4 Jan., the second of the twin Grail spacecraft moving into orbit around the Moon on the 2nd of Jan., and the proximity of the Moon and the planet Jupiter in the night sky on the 2nd of Jan. Searches excluding the event related terms were also conducted (see Table 4) as non-overlapping sets in order to assess the significance of results.

Table 4. Terms used in modified searches, refer to table 2 for details of original searches

Search label	**Terms**
Space AND grail	Space AND gear AND (grail|lunar|moon)
Space NOT grail	Space AND gear AND NOT (grail|lunar|moon)
Space AND jupiter	Space AND bodies+ AND (jupiter AND moon)
Space NOT jupiter	Space AND bodies+ AND NOT (jupiter AND moon)
Astro AND quad	Astro AND (quadrantid|meteor shower)
Astro NOT quad	Astro AND NOT (quadrantid|meteor shower)

Naïve interpretation of the results in Figure 2 (left) now seems promising. There is a low 1hAF for the Space and Grail search in the third quarter of the day (@18, 0.23), which contains retweets of posts about Grail tweeted by NASA and SETI in the first half of the (USA) working day. 1hAF for Space AND jupiter is generally high, and particularly in the third quarter (@18, 0.83), with retweets typically of links to pictures taken the previous night. Finally, the 1hAF values for Astro and quad are in the range 0.11-0.22, some of the lowest we saw, compared to between 0.32 and 0.39 for Astro AND NOT quad. These retweets are of messages from several sources reminding people to get up before dawn in order to see the meteors.

However, the differences in sample size between the searches about events and the exclusion searches we are using for comparison could be extreme: the largest (Astro NOT quad @24) contains 26327 retweets, the smallest (Space and jupiter @18) which generates 1hAF of 0.83 contains just 6 retweets. We therefore require a method of determining whether the 1hAF values we are seeing are significant or are merely the effect of small samples. To gain insight into the significance of results we used a funnel plot (see Figure 2 right). Funnel plots are employed in meta-analyses to detect publication bias and other biases [18]. The rationale is that small samples are expected to show higher variance. Therefore, if the measured values (in this case 1hAF) are plotted on the x axis and sample size on the y axis, then if all the data points come from the same population a triangular spread of points around the mean would be expected. Asymmetry in the plot suggests the data points may not all come from the same population. Figure 2 (right) presents a funnel plot for the events searches, excluding the points for Astro NOT quad, which have retweets in the thousands (plotting these would force us to use a log scale for the y axis making the funnel plot much harder to interpret).

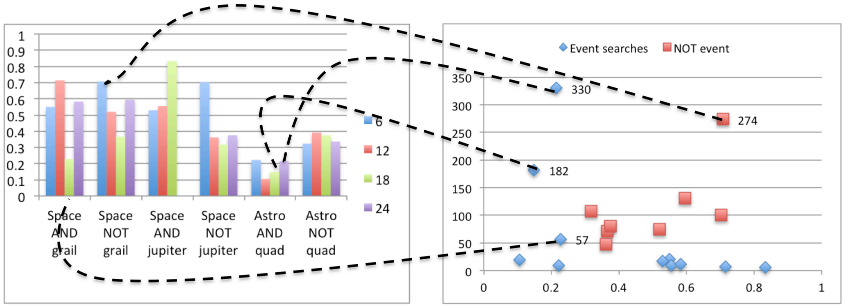


Fig. 2. Left: Ageing Factor at one hour, 1hAF, for searches modified to detect specific astronomical events, measured for 4*6 hour periods over the course of 3rd Jan. 2012. Right: Funnel plot of 1hAF vs number of retweets for the same data (excluding Astro NOT quad).

Based on the funnel plot, two of the Astro AND quad points still look interesting (@time 1hAF RT: @18 0.15 182, @24 0.22 330). Therefore, we can infer that retweeting activity around the Quadrantid meteor shower was significant in the hours of darkness for the UK and USA, where the largest concentrations of English speaking Twitter users in the northern hemisphere would be expected to be. Other data points which previously looked interesting, such as (@18 0.23 57) for Space and grail, on the funnel plot look like normal fluctuation within the expected variance of the population. This suggests that, at least on the 3rd of Jan 2012, the progress of Grail was not exciting the public to a significant degree.

A third data point also deserves investigation, that for Space NOT grail (@6 0.71 274). Examination showed that 216 of the retweets in this set contained the phrase "*join NASA*", from retweets of the humorous posts we identified earlier. We infer that people had a long attention span for that particular joke.

3 Discussion and Future Directions

Our objective in carrying out the two exploratory experiments described here was not to rate jokes, but to test ageing factor as an altmetric for analysing scientific communication in social media, and specifically to test whether it can give any insights for the smaller datasets typical of scientific content. For ageing factor calculated with a window of one hour (1hAF), several of our naïve observations fitted both assumption 1 (ageing factors for topics which concern special or exciting events will be lower than suitable baselines) and assumption 2 (ageing factors which are higher than suitable baselines are associated with topics in which interest is sustained over time). However, when a funnel plot was used to identify data points which lay outside the area of expected variance, only three data points appear significant: two with low 1hAF for retweets about the Quadrantid meteor shower, and one with high 1hAF for a (non-scientific) humorous post. We conclude that further investigation of 1hAF would be worthwhile, but that interpretation of the metric without reference to sample size must be avoided.

In these exploratory experiments, topics were identified using SQL searches. It would be intriguing to compare topics with low and high AF to the statistical linguistics approach used by Hu et al. [19] to classify tweets into *episodic* and *steady* categories. Hu defines episodic tweets as "*tweets that respond specifically to the content in the segments of the events*" and steady tweets as those *"that respond generally about the events"*. Our intuitions about how ageing factor works suggest that episodic tweets would be more likely to show low 1hAF and steady tweets more likely to show high 1hAF.

24hAF gave similar values for all the searches in experiment 1. Therefore, we did not use 24hAF in experiment 2. As in [10], when analysing twitter data, especially about a specific event or topic, there is an upper limit beyond which relevant tweets tail off. In our experiments this was 24 hours, in [10], which examined scientific publications, with a significantly longer lifetime, this started at 7 days, and up to 10, for publications released within a three month window. Future studies will look at a wider range of time windows to see if they give more sensitive results than the 24h window and will apply the funnel plot technique to check significance. For example, a six hour window (6hAF) might be interesting to observe for studies like experiment 2 which divide the day into quarters.

The overall aim of this work is to contribute to the nascent development of methods and metrics that will support analysis of public online scientific communications. It is clear that the big issue in achieving this is the level of noise in samples coupled with low actual levels of scientific communication in social media. These combine to make it difficult to get big enough samples to get statistically significant results. As an additional problem, the usage of terms on Twitter clearly varies considerably even over a few weeks: our experiments used data collected only a few weeks apart, but the searches developed in experiment 1 proved useless in experiment 2. This may make it difficult to devise standard filters for on-going monitoring of scientific communication. Noise was addressed in this study by writing SQL queries to produce disambiguated subsets. However, in the future we will need to identify, and possibly develop, more subtle, NLP-based techniques for classifying tweets on science related topics. These techniques will need to adjust dynamically to pick up new topics as they arise.

As for future work, although our interest in ageing factor was stimulated by the small sample sizes we found for typical scientific topics, we are investigating the application of the technique to larger datasets and longer sequences of events. Furthermore, we have not explored the differences between types of participants. For example, is there a difference between ageing factors observed for private individuals' tweets vs professional scientists' vs organizations'? Techniques for distinguishing these groups will be particularly important in achieving our overall goal of analysing public opinion about science.

4 Acknowledgements

A-S Dadzie is currently funded by the UK MRC project Time to Change (129941). Thanks to Mike Thelwall and the Statistical Cybermetrics Research Group for encouragement with pursuing the informetrics approach, and to Gary Simpson at Aston University for advice on the statistical issues.

5 References

1. M. Thelwall, "Webometrics," in *Annual Review of Information Science and Technology*, 2005, pp. 81-135.
2. J. Bar-Ilan, S. Haustein, I. Peters, J. Priem, H. Shema, and J. Terliesner, "Beyond citations: Scholars' visibility on the social Web," 2012.
3. J. Priem, H. A. Piwowar, and B. M. Hemminger, "Altmetrics in the wild: Using social media to explore scholarly impact," *arXiv12034745v1 csDL 20 Mar 2012*, vol. 1203.4745, pp. 1-23, 2012.
4. D. Ponte and J. Simon, "Scholarly Communication 2.0: Exploring Researchers' Opinions on Web 2.0 for Scientific Knowledge Creation, Evaluation and Dissemination," *Serials Review*, vol. 37, no. 3, pp. 149-156, 2011.
5. J. Letierce, A. Passant, J. Breslin, and S. Decker, "Understanding how Twitter is used to widely spread Scientific Messages," in *In Proceedings of the WebSci10 Extending the Frontiers of Society OnLine*, 2010.
6. K. Weller, E. Dröge, and C. Puschmann, "Citation Analysis in Twitter. Approaches for Defining and Measuring Information Flows within Tweets during Scientific Conferences," in *MSM2011 · 1st Workshop on Making Sense of Microposts*, 2011, no. May, pp. 1-12.
7. A. Mandavilli, "Trial by Twitter," *Nature*, vol. 469, pp. 286-287, 2011.
8. D. A. Nisbet, Matthew C., Scheufele, "What's new for science communication? Promising directions and lingering distractions," *American Journal of Botany*, vol. 10, pp. 1767-1778, 2009.
9. J. Clark and P. Aufderheide, "Public Media 2.0: Dynamic, Engaged Publics," Center for Social Media, 2009.
10. G. Eysenbach, "Can tweets predict citations? Metrics of social impact based on Twitter and correlation with traditional metrics of scientific impact," *Journal of Medical Internet Research*, vol. 13, no. 4, p. e123, 2011.
11. A. Hubmann-Haidvogel, A. M. P. Brasoveanu, A. Scharl, M. Sabou, and S. Gindl, "Visualizing Contextual and Dynamic Features of Micropost Streams," in *Proceedings of the WWW'12 Workshop on "Making Sense of Microposts" , , April 16, 2012. CEUR Workshop Proceedings V838*, 2012.
12. C. Chew and G. Eysenbach, "Pandemics in the Age of Twitter: Content Analysis of Tweets during the 2009 H1N1 Outbreak," *PLoS ONE*, vol. 5, no. 11, p. 13, 2010.

13. V. Uren and A.-S. Dadzie, “Relative Trends in Scientific Terms on Twitter,” in *altmetrics11: Tracking scholarly interest on the Web, Workshop at ACM WebSci’11*, 2011.
14. K. Weller and C. Puschmann, “Twitter for Scientific Communication: How Can Citations/References be Identified and Measured?” in *Proceedings of the ACM WebSci11*, 2011, pp. 1-4.
15. K. Weller, E. Dröge, and C. Puschmann, “Citation Analysis in Twitter: Approaches for Defining and Measuring Information Flows within Tweets during Scientific Conferences,” in *Proceedings of the WWW’12 Workshop on “Making Sense of Microposts” , , April 16, 2012. CEUR Workshop Proceedings V838*, 2012.
16. L. Egghe and R. Rousseau, *Introduction to Informetrics*. Elsevier, 1990.
17. Y. Mejova and P. Srinivasan, “Crossing Media Streams with Sentiment: Domain Adaptation in Blogs, Reviews and Twitter,” in *Sixth International AAAI Conference on Weblogs and Social Media*, 2012, pp. 234-241.
18. M. E. Egger, G. Davey Smith, M. Schneider, and C. Minder, “Bias in meta-analysis detected by a simple, graphical test,” *British Medical Journal*, 1997.
19. Y. Hu, A. John, D.´e D. Seligmann, and F. Wang, “What Were the Tweets About? Topical Associations between Public Events and Twitter Feeds,” in *Sixth International AAAI Conference on Weblogs and Social Media*, 2012, pp. 154-161.
20. J. D. Miller, "Public Understanding of, and Attitudes toward, Scientific Research: What We Know and What We Need to Know", in *Public Understanding of Science*, v. 13, 2004, pp. 273-294.

Lifting user generated comments to SIOC

Julien Subercaze and Christophe Gravier

LT2C, Télécom Saint-Étienne, Université Jean Monnet
10 rue Tréfilerie, F-4200 France
{julien.subercaze,christophe.gravier}
@univ-st-etienne.fr
http://portail.univ-st-etienne.fr

Abstract. HTML boilerplate code is used on webpages as presentation directives for a browser to display data to a human end user. For machines, our community has made tremenduous efforts to provide querying endpoints using agreed-upon schemas, protocols, and principles since the avent of the Semantic Web. These data lifting efforts have been some of the primary materials for bootstrapping the Web of data. Data lifting usually involves an original data structure from which the semantic architect has to produce a mapper to RDF vocabularies. Less efforts are made in order to lift data produced by a Web mining process, due to the difficulty to provide an efficient and scalable solution. Nonetheless, the Web of documents is mainly composed of natural language twisted in HTML boilerplate code, and few data schemas can be mapped into RDF. In this paper, we present CommentsLifter, a system that is able to lift SIOC data from user-generated comments in Web 2.0.

Keywords: Data extraction, Frequent subtree mining, users comments

1 Introduction

The SIOC ontology [5] has been defined to represent user social interaction on the web. It aims at interconnecting online communities. Nowadays SIOC data are mostly produced by exporters [1]. These exporters are plugins to existing frameworks such as blog platforms (Wordpress, DotClear , . . .), content management systems (Drupal) and bulletin boards. Unfortunately, these exporters are not yet default installation plugins for these frameworks, except for Drupal 7. Therefore few administrators enable them and as a consequence the SIOC data production remains rare[2]. There exists also numerous closed source platforms that supports online communities. Among them are the online newspapers that allow commenting on articles, Q&A systems. This subset of the user generated content on the web will never be unlocked using exporters.

In this paper we present CommentsLifter, a web-mining approach that aims at extracting users' comments directly from HTML pages, in order to circumvent

[1] http://sioc-project.org/exporters

[2] http://www.w3.org/wiki/SIOC/EnabledSites

the exporter issue. The comments are identified in webpages by mining frequent induced subtrees from the DOM , and using heuristics allow to discriminate the different field of the comment (username, date, ...). This approach does not require any a priori knowledge of the webpage. We empirically evaluated our approach and obtained very good results.

The paper is structured as follows. The next section presents related works on both structuring data into semantic web formats and web mining approaches. Section 3 presents a formalisation of the problem and recalls some theoretical tree mining results. Section 4 details the different steps of CommentsLifter, followed by experimental results. Finally, section 6 concludes.

2 Related works

Converting existing format of data into RDF is a cornerstone in the success of the semantic web. The W3C maintains a list of available RDFizers on its website [3]. Input data can be either structured or unstructured. In the former case, if the semantics of the data can be extracted, then the conversion can take place without human intervention [14], otherwise the user needs to manually specifiy the semantics of the data. Sesame contains an API called SAIL (Storage And Inference Layer) that can be used to wrap existing data format into RDF. The BIO2RDF project [3] uses this API to build bioinformatics knowledge systems. Van Assem presented a method for converting Thesauri to RDF/OWL [2] that has been successfully applied for biological databases [3]. In order to convert a mailing list archive into a RDF format, the authors of [11] developed SWAML, a python script that reads a collection of messages in a mailbox and generates a RDF description based on the SIOC ontology. Since the input data are already structured (i.e. emails follow the RFC 4155) the conversion is straightforward. On the other side, there exist several approaches that aims at automatically or semi-automatically adressing the case where user intervention is usually required. Text-To-Onto [19] is a framework to learn an ontology from text using text mining techniques as well as its successor Text2Onto [9]. However none of these papers provide a sound evaluation of the quality of learnt ontologies. This is due to the very nature of ontology modeling in which no ground truth can be assessed, as there exists as many models as one could imagine for describing the same thing. In [4], Berendt details relationships between web mining and semantic web mining. The different cases (ontology learning, mapping, mining the semantic web, ...) are detailed. From this categorization, the purpose of our research falls into the category of *instance mining*, which focuses on populating instances for existing semantics. For this purpose, learning techniques have been proposed for web scale extraction with a few semantic concepts [10] and presented promising results at the time of publication. Textrunner [22] also learns to perform web-scale information extraction, presenting good precision but a very low recall. Concerning non learning techniques, automatic modelling of user profiles has

[3] `http://www.w3.org/wiki/ConverterToRdf`

been performed in [12], using term recognition and OpenCalais for named entity recognition.

Several techniques have been developed for web extraction. In [6], the authors simulate how a user visually understands Web layout structure. [17] divided the page based on the type of tags. Many recent research works exploit text density to extract content on pages [16, 21, 15]. This approach presents good results regarding article content extraction. In order to do so, the boilerpipe library[4], based on the work from Kohlschutter [16, 15] is widely used. For a more detailed survey on the different Web data extraction techniques we encourage the reader to refer to [7]. Among other techniques DEPTA [23] (an extension of works done in [18]) presents a hybrid approach of visual and tree analysis. It uses a tag tree alignment algorithm combined with visual information. In a first step DEPTA processes the page using a rendering engine (Internet Explorer) to get the boundaries information of each HTML element. Then the algorithm detects rectangles that are contained in another rectangle, and thus build a tag tree in which the parent relationships indicates a containment in the rendered page. DEPTA then uses a string edit distance to cluster similar nodes into regions. Since each data region in a page contains multiple data records, extracted tag trees must be aligned to produce a coherent database table. A tree edit distance (like in [20]) is then defined and used to merge trees. However DEPTA is not able to extract nested comments. We will use a different approach, that only requires a DOM parsing technique and that is suitable for analyzing huge amount of pages. Our approach is based on a theoretical tree mining background, presented in the next section.

3 Problem Definition

The purpose of our work is to provide a solution for the leverage of Linked Data using the SIOC schema from user generated comments on webpages, without any a priori knowledge on the webpage. Our main assumption is that comments on a given webpage (even at website scale) are embedded in the same HTML pattern. This assumption is well fulfilled in practice since comments are usually stored in a relational database and exposed into HTML after an automatic processing step. Therefore our goal is to automatically determine the HTML pattern that is used to expose the comments and then to identify the relevant information in the content to fill SIOC instances.

Basically a comment is a *sioc:Post* contained in a *sioc:Forum* container. We identified the following subset of the core-ontology properties of *sioc:Post* to be relevant for the extraction (we marked with * the mandatory properties and relationships) :

***sioc:content**:** text-only representation of the content
***dc:terms*:** title of the comment
***dcterms:created*:** creation date

[4] http://code.google.com/p/boilerpipe/

and relationships :

***sioc:has_creator**:** points to a *sioc:UserAccount* which is the resource
***sioc:has_container**:** indicates a *sioc:Container* object
***sioc:reply_to*:** links to a *sioc:Post* item
***sioc:has_reply*:** links to *sioc:Post* items

The *sioc:Container* pointed by *sioc:has_container* can be from different types in our case. We consider extraction from user reviews (*sioct:ReviewArea*), posts on forum (*sioc:Forum*), comments from blogs (*sioct:Weblog*), Q&A answers (*sioct:*) or generally for newspaper discussion (*sioc:Thread*). However, distinguishing these differents subclasses of *sioc:Container* would require classification from the webpages we intend to perform extraction on. This is out of the scope of our paper, we will therefore uniformally consider the container as an instance of *sioc:Container*. Similarly there exists subclasses of a *sioc:Post* for each container. As for the container, our algorithm will output *sioc:Post* items.

To summarize, our problem is the following : in order to generate SIOC data from raw HTML, we must identify the different items (*sioc:Post*) and their conversational relationships (*sioc:reply_to* and *sioc:has_reply*). For each item we must identify the user (*sioc:UserAccount*), the content of the post (*sioc:content*) and when possible the date and the title.

3.1 Frequent Subtree Mining

In the case of product listing extraction, the goal is to extract frequent subtrees that are identical in the page. For this purpose, a bottom-up subtree mining objective is sufficient. For selecting a feature among mined items, for example title and price, a filter on the leaf nodes can be applied. In the case of comments extraction, the patterns can be nested. Assuming that the pattern we are looking for is $[a[p; br; p; div]]$, if we encounter a reply to a comment, i.e. nested instances in the pattern, the tag tree for the comment and its answer could be as follows : $[a[p; br; p; div; [a[p; br; p; div]]]]$. In the case of a single comment we will encounter our pattern $[a[p; br; p; div]]$. We observe that nodes can be skipped horizontally along with their descendants.

In frequent subtree mining, three types of subtrees are distinguished : *bottom-up*, *induced* and *embedded*. Figure 1 depicts these different subtrees. For more details on frequent subtree mining, we refer the reader to [8].

We observed empirically that instances are nested in the way we described previously : the direct parenting relation is preserved. Consequently *induced subtree* is a sufficient type of subtree for our purpose. *Embedded subtree* mining could also be used, however since the algorithms complexity grow with the complexity of the pattern to mine, *induced subtree* mining is definitely more appropriate. The main advantage of subtree mining over existing works is that it provides a *mine once extract many* approach. In order to mine large websites, one would need to mine the pattern from only one webpage from each site and could later extract data by simple pattern matching on other pages, thus

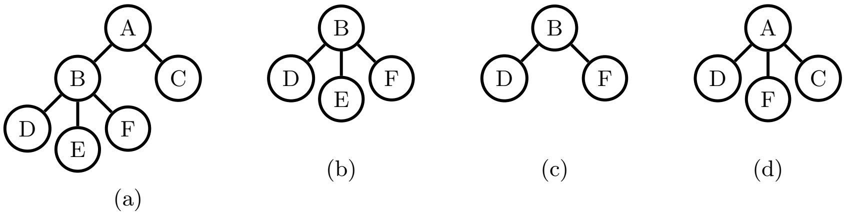


Fig. 1. Different types of subtree from tree (a) : bottom-up subtree (b) ; induced subtree (c) ; embedded subtree (d)

saving large amount of computational resources and time. In the next section, we present CommentsLifter, our approach to comments' extraction based on frequent subtree mining.

4 CommentsLifter

We now start the description of CommentsLifter. This section presents the different steps of our algorithm. The underlying idea of CommentsLifter is to use simple observations on Web page structures to reduce the candidate set generation. This allows us to minimize the error rate while selecting the winning pattern, and to contribute to the runtime performance optimization objective. To extract comments from a Web page, CommentsLifter uses seven steps: Document preprocessing, Frequent subtrees extraction, Clustering, Merging, Pattern expansion, Winner selection, Field extraction, Data extraction. In the next subsections we detail the process followed in each step of our algorithm.

4.1 Preprocessing

Recommendations issued by the W3C aim at specifying the languages of the World Wide Web, with various versions of HTML and CSS. While pages following these recommendations produce clean DOM trees, they represent only 4.13 % of the Web pages[5]. The remaining pages are made up of wrongly formatted HTML code that is often referred to as "tag soup"[6]. Due to the large portion that these pages represent, Web browser engines are able to handle malformed HTML tags, improperly nested elements, and other common mistakes. In our algorithm, the first step is to convert any input HTML document (malformed or not) and to output a well-formed document as a DOM tree. For this purpose, any dedicated library (Jsoup[7], Jtidy[8]) or browser engine can be used.

[5] http://arstechnica.com/Web/news/2008/10/opera-study-only-4-13-of-the-Web-is-standards-compliant.ars

[6] http://en.wikipedia.org/wiki/Tag_soup

[7] http://jsoup.org/

[8] http://jtidy.sourceforge.net/

4.2 Frequent subtrees extraction

The next step of our algorithm consists of the generation of a first candidate set. For this purpose we extract frequent depth-two trees from the DOM and store them in a cardinality map (an example is given in Table 1). The basis of our approach is to select patterns with a depth of two that will be expanded into larger patterns (empirical results in Section 5 show that the average depth of a comment pattern is 4.58). For this purpose, a tag tree is generated from the preprocessed DOM. CommentsLifter traverses the tree in a top down fashion and stores the encountered trees of size two (each node that has children, along with its children). The results are sorted, as presented in Table 1. Candidates with less than two occurrences are discarded, since we assume that there are at least two comments. The same assumption is made for instance by [23]. For each pattern occurrence, the encountered instance is stored in a multimap (in fact we only store the label of the parent node).

Count	13	12	10	8
Tree	div[a;br;i;p]	ul[li;li]	div[p;p]	div[p;p;p]

Table 1. Example of two depth candidates.

At this stage our candidate set is initialized and contains the instances of comments we are looking for.

4.3 Clustering

Comments in a Web page appear in a continuous manner, so it is very unlikely that comments are stored in different branches of the DOM tree. We did not encounter the case of split comments in different subtrees during our experiments. In fact, comments are organized in a tree structure, where the beginning of the comments block is the root of the tree. Since comments are located in the same subtree, we proceed to a clustering phase of the occurrences for each pattern.

In this step, we aim at clustering co-located instances of the same pattern. In other words, the algorithm builds the pairs $(pattern, Instances)$, where each pattern is associated with a set of instances matching it that are located in the same subtree and close to each other. Consequently, one pattern can be associated multiple times with a unique set of instances, whose member is distinct from any other member of another associated set to the pattern. This means that each pair $(pattern, Instances)$ is splitted into different $(pattern, Instances)$ where the instances in $Instances$ do belong to the same subtree in the DOM. The basis of our algorithm is a distance-based clustering.For each given pattern, the algorithm sorts its instances along their depth in the tree. At each depth, we check for each instance if it has a parent in the previously found $(pattern, Instances)$. For the remaining instances, we cluster them using a classical node distance in trees : $d(a, b) \in \mathbb{N}$ is the length of the shortest path between the nodes a and *b*. After building the set of $(pattern, Instances)$, we remove elements where the cardinality of instances is equal to one.

For example, running our algorithm on the tree provided by Figure 2 with the pattern $ul[li; li]$ would produce two pairs $(pattern, Instances)$ that are depicted with dotted and dashed boxes in the same figure.

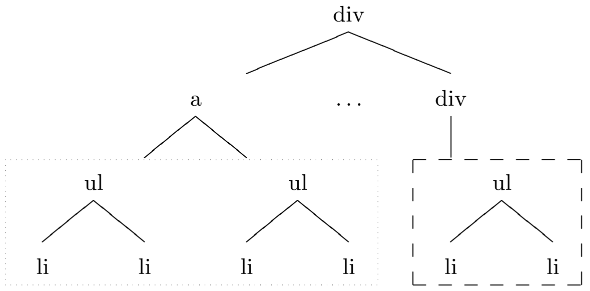


Fig. 2. Example output for section 4.3 with the pattern $ul[li; li]$

At the end of this step, CommentsLifter holds the set of pairs $(pattern, Instances)$ that matches a pattern to a set of co-located instances in the DOM tree. In the next step, we try to identify mergeable patterns from this data structure.

4.4 Merging

HTML patterns that contain comments often have a depth greater than two (see Section 5 for more details). Consequently our candidate set may at this step contain pairs of $(pattern, Instances)$ that belong to the same global pattern, that we did not discover yet because the pattern expansion process will occur later. Instead of discovering the same pattern from different candidates, we aim at merging these candidates beforehand for optimization.

For this purpose we perform a pairwise comparison between the sets of instances. We process a merge between sets of instance when every element of a set S_1 is topped by an element from the other set of instances. An element is topped by another if the latter is a parent of the former. If a set of instances is topped by another set, we discard this set and restart the process until no further updates to the candidate set are performed.

Algorithm 1 presents this merging process. After this step, the candidate set is again drastically pruned since we eliminated all potentially duplicate patterns for the expansion phase. This process, together with the previous clustering process, are the key phases of our algorithm since they discard both duplicate and irrelevant candidates.

Algorithm 1: *MergePatterns* : Merge similar pattern

```
Data: A collection of couple (Pattern,Instances) $Input$ where all patterns are
      different,
$minDepth$ the mininum depth of the instances in the DOM tree
Result: A collection $Candidates$ of (pattern,Instances) where similar couples
        have been merged
$recursion \leftarrow false$;
$Candidates \leftarrow \emptyset$;
for $i : 0 \ldots Input.size$ do
    for $j : i \ldots Input.size$ do
        //Depth of the highest common element ;
        $commonDepth \leftarrow depth(HighestCommon(Input[i].Instances,$;
        $Input[j].Instances))$;
        //Save useless computation ;
        if $commonDepth < minDepth$ then
            continue ;
        end
        $Small \leftarrow setWithLessInstances(Input[i], Input[j])$;
        $Large \leftarrow setWithMoreInstances(Input[i], Input[j])$;
        if $\forall k \in Small.Instances, \exists l \in Large.Instances, isParentOf(l, k)$ then
        $Candidates \leftarrow Input \setminus Small$;
        //Restart the merging process;
        $MergePatterns(Candidates)$;
        //Cut the current call;
        $break$;;
        else $Candidates \leftarrow Candidates \cup Large \cup Small$
    end
end
$return$ $Candidates$;
```

4.5 Pattern expansion

Since the candidate set contains simple patterns (i.e. of depth two), we process a pattern expansion to discover the fully matching patterns. Patterns may be expanded in both directions, towards the top and the bottom of the tree.

For each candidate $(pattern, Instances)$, we distinguish two cases. In the first case every instance is at the same depth in the tree, this is the case of product listing extraction that we call the *flat case*. This is the case of bottom-up subtree mining (see Figure 1). The second case also called the *nested case* is the one where instances belong to the same subtree but at different depths, this is the case of induced subtrees in Figure 1 . Top expansion is straightforward, we check if the type of the parent node (i.e. HTML markup tags) for every instance is the same, in this case we expand the pattern with the new parent node and update the instances consequently. This process is executed until all the instances do not share the same tag as parent node or if the same node (in sense of label in the tree, not HTML markup tag) is the parent of all instances. This process is the same for both *nested* and *flat* cases.

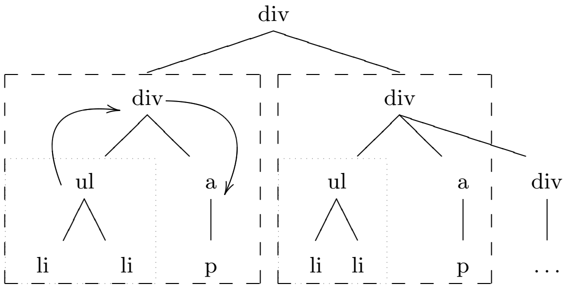


Fig. 3. Pattern expansion process, top expansion until the same node (top div node) is shared by both instances, then bottom expansion for induced subtree (the most right div is skipped)

Once a pattern is expanded in the top direction, the bottom expansion takes place. This bottom expansion in the *flat case*, similar to the top expansion is simple since *top-down subtrees* are easy to extract. One just needs to traverse the instances trees in a top-down lefmost direction, looking for nodes that are shared by all instances. Once a node is not present in every instance the algorithm uses backtracking to select the next sibling, and then continues its process.

However the *nested case* is not trivial since we look for embedded subtrees. Standard algorithms such as AMIOT [13] and FREQT [1] are dedicated to this task, but as we mentioned in section 3.1 they performed poorly on the full Web page, either the running was excessively long (some minutes) or the program ended with an exception. To make efficient use of these algorithms, we take advantage of the specificities of comments extraction. Apart from the aforementioned issues, the biggest problem we encountered using AMIOT or FREQT was that we were unable to adaptatively select the support (in fact, a percentage) for a Web page. However in our pattern expansion case, we know how much instances are present for each couple $(pattern, Instances)$. Thus, we performed a modification in the algorithms (see Implementation in section 5.1 for further details) to discard candidate patterns not on a minimum support base, but on a strict occurrence equality base. Finally, instead of using the complete Web page as the data tree, we construct a tree by adding all the instances in a tree. More precisely since the instances are stored using only their top nodes (the whole instance is retrieved by applying the pattern to this node in the data tree), we build the data tree by adding the subtrees under the instances' top nodes as children of the root node. Therefore, our data tree contains on ly the relevant instances. Consequently the set of candidate patterns is drastically reduced, compared with the use of the whole DOM page. To drive AMIOT in the right direction, the candidate pattern set is initialized with the current depth-two pattern, thus avoiding useless candidate generation for the first stage. Figure 3 depicts this expansion process. In this figure we present the instances as they appear in the DOM tree. Our starting pattern is $ul[li; li]$, and its instances are represented within the dotted boxes. Both instances have a div node with a different label as parent, consequently the pattern is expanded to the top : $div[ul[li; li]]$. Next, both instances again have a div node as parent but in this case this is the same node. The top expansion process finishes. For the bottom expansion, we consider this subtree as the data tree. AMIOT (resp. FREQT) performs a left to right expansion that adds the node a to the pattern: $div[ul[li; li]; a]$. The rightmost div node is discarded since it does not belong with the left instance (the occurrence is one whereas the algorithm expects two). Then AMIOT adds the node p to the pattern that becomes $div[ul[li; li]; a[p]]$. The figure does not show the part under the right most div, in this part we could find for example another instance of $div[ul[li; li]; a[p]]$, resulting in a nested comment.

4.6 Winner selection

Recurring structures competing with the comments pattern in the candidate set are usually menu elements, links to other articles. In [15, 16], Kohlschütter developed a densitometric approach based on simple text features that presents excellent results (F_1-Score : 95 %) for news article boilerplate removal. User generated comments also differentiate from menu elements on their text features. Comments are from different text lengths, the link density is low since comments are not part of a link in comparison to menu items. We developed simple heuristics, based on our observations, to discard irrelevant candidates and to rank the remaining candidates.

Our experimentation showed that instances with a link density greater than 0.5 (Kohlschutter found 0.33 for news article) are always boilerplate. Short comments in very complex HTML boilerplate patterns can produce instances with a quite high link density. We empirically observed that links are on the username or on its avatar and link to a profile page. Consequently we discard candidates where the average link density is above 0.5.

For the remaining candidates their score is given by the following formula:

$$Score(p, I) = \overline{lgt(I)} \times \sqrt{\frac{1}{n-1}\sum_{j=1}^{n}\overline{lgt(I)} - lgt(I_j)}^{2} \quad (1)$$

The above formula computes the average text length times its standard deviation. This heuristic promotes candidates where the instances have longer text length with variable length. Finally we did not use heuristics based on a lower bound of words (as in [15, 16]). Once again comments extraction differs from traditional boilerplate removal since some comments are sometimes just one word (e.g. *lol*, *+1*, *first*) or they may be longer than the article they are commenting on. Therefore the standard deviation is very useful for eliminating menu items where the text length is often very close among their instances. The $(pattern, Instances)$ couple with the highest score is promoted as the winner. At this step, the algorithm output the tree of instances, i.e. we have the structure of the conversation, but we need to further structure the conversation by identifying the different fields in the pattern.

4.7 Structuring the content

Once the HTML pattern containing comments has been selected, the next step of our algorithm consists of extracting the related SIOC fields as we described in section 3. Two fields are always present in a comment : the username and the content of the comment. From our observations on various websites (online press, Q&A, blogs, reviews), two other fields appear often. Firstly the date when the comment has been posted occurs in 97.95 % of the cases (See Table 2). Comments less often have a title, but the percentage we measured (24.48 %) remains high enough to be of interest. We voluntarily skipped the extraction of rare fields (vote on comments, account creation date) because of their few occurrences and the fact that they complicate the whole field's identification process. From our observations, we noticed that content and title are always contained in their own HTML markup tags, i.e. it is very unlikely to see the title and the content in the same `div`. However we noticed that username and date often occur between the same markup tag, for example we often encountered comments fields such as `<div>John, May 25, 2012 at 5:00 p.m</div>`. Therefore we apply the following procedures : first, identifying the date field within the pattern. We store the location of the date in the pattern and remove dates in every instances. For instance, the previous example will become`<div>John</div>`. At this point we are sure that the fields we are looking for will be in distinct nodes of the pattern.

Date parsing Date parsing library such as JodaTime[9] and JChronic[10] perform well on extracting date from messy data. These libraries takes a String as input and return a Date object. However they do not offer the feature of returning the original text without the substring containing the date. Therefore we developed our library, using features and heuristics from both of the above mentioned libraries, that offers the date string removal feature.

Fields selection Our heuristic is based again on simple observations. We know that every comment contains at least a username and a content, with date and title being optional. For each leaf of our pattern, we compute the following measures over its instances : percentage of date found, average text length, standard deviation of text length, text entropy and average word count, standard deviation on word count. Using these values, we build two candidate sets, the first one for the date, and the other one for title, content and username. We distinguish these fields using the fact that title, content and username are unstructured text, however dates have a particular structure (containing year, days, ...).

A node in the pattern is a candidate for the date field if its percentage of date found is over 0.7 (to take into account the fact that date extraction is not perfect), has more than two words and has a coefficient of variation on the word count that is inferior to 0.2. This latter condition requires that the number of words is very close from one instance to another. We did not set the value to zero to avoid discarding fields where the date has a variable length, for example `one hour ago` and `yesterday`. From these candidates, we pick the field with the highest entropy in order to discard constant fields that may have been recognized as a date. However if the set is empty, then no dates are specified for the comments, in practice this happens very rarely.

The second candidate set should contain only nodes that instances own textual data. For this purpose we discard the nodes in the pattern where the variation of text entropy is equal to zero (constant text in every instance). Since we know that the node containing the content must be present within this set, we aim at identifying this field in the first place. Luckily the content is very simple to identify since it contains the most words, has a very variable length and word count. In the practice, selecting the node having the highest word count average is sufficient. Once the content has been removed from this candidate set, we first check its size. If the size is equal to one, the remaining candidate matches the username. In the case where two candidates are present, we have to distinguish between username and title. Usernames are very short names, usually one or two words, and are then in average shorter than the title. If two fields are present, the shorter is identified as being the username and the longest is then the title.

[9] `http://joda-time.sourceforge.net/`

[10] `https://github.com/samtingleff/jchronic`

		Ground	**True**	**False**
	Pages	**Truth**	**Positive**	**Positive**
Global	100	2323	2121	153
Flat	77	1837	1674	125
Nested	23	486	447	26

	Precision	**Recall**	F_1
Global	93,3 %	91,3 %	92,3 %
Flat	93.1 %	91.1 %	92.1 %
Nested	94.5 %	91.97 %	93.22 %

Table 2. Evaluation : steps one to six

	Content	**Username**	**Date**	**Title**
Occurrence (%)	100	100	97.59	24.48
Correct extraction (%)	100	87.75	83.67	81.63

Table 3. Evaluation, step seven : field identification

5 Mining experiments

This section presents the evaluation protocol of CommentsLifter as well as our experiment's results. We first detail the experimental setup, which is a bit particuliar for comments extraction due to the large use of AJAX. Finally we present global and detailed results for both flat and nested cases.

5.1 Setup

Many Web pages handle comments using AJAX, consequently downloading raw HTML along manual ground truth construction is not sufficient for building the dataset. To circumvent this issue we developed two components:

Firefox Extension We implemented a Firefox extension that sends the current DOM (after browser-side Javascript processing) to a Web server.

Web Server The server receives the DOM from the browser through a POST request on a servlet, then runs CommentsLifter and presents an evaluation form along with the extracted comments. The user is asked to evaluate the pertinence of the extraction. We distinguish three cases for the result. We used Jena[11] to generate the SIOC output.

5.2 Evaluation

The results obtained from our evaluation are given in Table 2 and Table 3. Table 2 presents extraction results of the pattern mining part (steps 1 to 6 of

[11] `http://jena.apache.org/`

the algorithm) whereas Table 3 presents the field identification results (step 7). The first six steps of the algorithm present very good results, with a global F_1 score over 92%. Concerning field identification we first present the occurrence of the different fields over our dataset. Username and content are always present, while the date is not far from being present in every comment. However titles are to be found in one quarter of the comments. The evaluation accords to the heuristics we describe in section 4.7. Content extraction is a straightforward task since it is easy to "measure" differences with other fields, our algorithm performs perfectly at this task. Date parsing is no easy task, however our algorithm still performs well with an identification rate of 83.67 %. However we note that while the rest of the process is language agnostic, date parsing libraries are designed to work with western languages (English, German, french, spanish, ...) but may fail with other languages, especially with non latin alphabet.

6 Conclusions and future work

In this paper, we presented CommentsLifter, an algorithm that extracts users' comments and outputs SIOC data. Our algorithm combines mining induced subtrees from the DOM with simple yet robust heuristics to select the pattern containing the comment as well as identifying several fields within the pattern. The empirical evaluation presents very good results, for both extraction and field identification. We successfully extracted comments from various types of Web sites, without *a priori* knowledge, such as online newspapers, forum, user reviews, blogs and we were able to reconstruct the conversations.

Further research will focus on refining the category of the extracted container, in order to determine whether the discussion takes into a forum, Q&A, blog or review area.

References

1. Tatsuya Asai, Kenji Abe, Shinji Kawasoe, Hiroshi Sakamoto, and Setsuo Arikawa. Efficient substructure discovery from large semi-structured data. pages 158–174, 2002.
2. Mark Van Assem, Maarten R. Menken, Guus Schreiber, Jan Wielemaker, and Bob Wielinga. A method for converting thesauri to rdf/owl. In *Proc. of the 3rd Intl Semantic Web Conf. (ISWC04), number 3298 in Lecture Notes in Computer Science*, pages 17–31. Springer-Verlag, 2004.
3. F. Belleau, M.A. Nolin, N. Tourigny, P. Rigault, and J. Morissette. Bio2rdf: towards a mashup to build bioinformatics knowledge systems. *Journal of biomedical informatics*, 41(5):706–716, 2008.
4. Bettina Berendt, Andreas Hotho, and Gerd Stumme. Towards semantic web mining. In *Intl Semantic Web Conf. (ISWC02)*, pages 264–278. Springer, 2002.
5. J. Breslin, A. Harth, U. Bojars, and S. Decker. Towards semantically-interlinked online communities. *The Semantic Web: Research and Applications*, pages 71–83, 2005.

6. D. Cai, S. Yu, J.R. Wen, and W.Y. Ma. Extracting content structure for web pages based on visual representation. In *Proc. of the 5th Asia-Pacific conference on Web technologies and applications*, pages 406–417. Springer-Verlag, 2003.
7. C.H. Chang, M. Kayed, R. Girgis, and K.F. Shaalan. A survey of web information extraction systems. *Knowledge and Data Engineering, IEEE Transactions on*, 18(10):1411–1428, 2006.
8. Y. Chi, R.R. Muntz, S. Nijssen, and J.N. Kok. Frequent subtree mining-an overview. *Fundamenta Informaticae*, 66(1-2):161, 2005.
9. P. Cimiano and J. Völker. Text2onto. *Natural Language Processing and Information Systems*, pages 257–271, 2005.
10. M. Craven, D. DiPasquo, D. Freitag, A. McCallum, T. Mitchell, K. Nigam, and S. Slattery. Learning to construct knowledge bases from the world wide web. *Artificial Intelligence*, 118(1):69–113, 2000.
11. S. Fernández, D. Berrueta, and J.E. Labra. Mailing lists meet the semantic web. In *Proc. of the Workshop on Social Aspects of the Web, Poznan, Poland*, 2007.
12. A. Gentile, V. Lanfranchi, S. Mazumdar, and F. Ciravegna. Extracting semantic user networks from informal communication exchanges. *The Semantic Web–ISWC 2011*, pages 209–224, 2011.
13. S. Hido and H. Kawano. Amiot: induced ordered tree mining in tree-structured databases. In *Data Mining, Fifth IEEE International Conference on*, 2005.
14. D. Huynh, S. Mazzocchi, and D. Karger. Piggy bank: Experience the semantic web inside your web browser. *The Semantic Web–ISWC 2005*, pages 413–430, 2005.
15. C. Kohlschütter, P. Fankhauser, and W. Nejdl. Boilerplate detection using shallow text features. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 441–450. ACM, 2010.
16. C. Kohlschütter and W. Nejdl. A densitometric approach to web page segmentation. In *Proceeding of the 17th ACM conference on Information and knowledge management*, pages 1173–1182. ACM, 2008.
17. S.H. Lin and J.M. Ho. Discovering informative content blocks from web documents. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 588–593. ACM, 2002.
18. B. Liu, R. Grossman, and Y. Zhai. Mining data records in web pages. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 601–606. ACM, 2003.
19. A. Maedche and S. Staab. Ontology learning for the semantic web. *Intelligent Systems, IEEE*, 16(2):72–79, 2001.
20. D.D.C. Reis, P.B. Golgher, A.S. Silva, and AF Laender. Automatic web news extraction using tree edit distance. In *Proceedings of the 13th international conference on World Wide Web*, pages 502–511. ACM, 2004.
21. F. Sun, D. Song, and L. Liao. Dom based content extraction via text density. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information*, pages 245–254. ACM, 2011.
22. A. Yates, M. Cafarella, M. Banko, O. Etzioni, M. Broadhead, and S. Soderland. Textrunner: open information extraction on the web. In *Proceedings of Human Language Technologies*, pages 25–26. ACL, 2007.
23. Y. Zhai and B. Liu. Web data extraction based on partial tree alignment. In *Proceedings of the 14th international conference on World Wide Web*, pages 76–85. ACM, 2005.

Comparing Metamap to MGrep as a Tool for Mapping Free Text to Formal Medical Lexicons

Samuel Alan Stewart*[1], Maia Elizabeth von Maltzahn[2], Syed Sibte Raza Abidi[1]

[1]NICHE Research Group, Dalhousie University, 6050 University Ave., Halifax, NS, Canada.
http://www.cs.dal.ca/ ~niche.
[2]Department of Internal Medicine, University of Saskatchewan, 103 Hospital Drive, Saskatoon, SK, Canada
*Contact Author: sam.stewart@dal.ca

Abstract. Metamap and Mgrep are natural language processing tools for mapping medical free text to formal medical lexicons, but an indepth comparison of the programs and their application to social media data has never been pursued. This project is interested in comparing the programs, in order to determine which program is most appropriate for mapping web 2.0 communication data. The archives of the Pediatric Pain Mailing List (PPML) were mapped with both programs, and each term returned was checked for correctness. The analysis resulted in Mgrep having a significantly higher precision (76.1% to 58.8%, difference of 18%, p-value < 0.0001) while Metamap returned more terms: 2381 to 1350. When considering only perfect or multiple matches, Mgrep still had better precision (81.2% to 71.3%, difference 10%, p-value < 0.0001). Ultimately Mgrep's precision may make it the better choice for many applications, but when there is more value in number of correct terms returned over accuracy of those terms, Metamap's larger set and superior scoring function may make it the tool of choice.

Keywords: Natural Language Processing; Semantic Mapping; MeSH; UMLS; Knowledge Management; Knowledge Translation

1 Introduction

Web 2.0 tools provide a valuable service to the healthcare community. Through online discussion forums, mailing lists, blogs, etc., clinicians can find mediums through which they can communicate their problems and share their experiences, developing relationships and creating a virtual community of practice (Wenger, 2004). Notwithstanding the evidence-based nature of modern healthcare, these online tools provide avenues for sharing experiential and tacit knowledge (Abidi, 2006) with colleagues in a way that spans the temporal and geographical boundaries that often prevent face-to-face communication.

The archives of these online conversations contain vast amounts of tacit and experiential knowledge. Extracting this knowledge and making it available to the community can improve the overall knowledge base, but how best to process this unstructured free text has proven a challenge.

Natural language processing approaches have been pursued in the past, including the semantic mapping of the unstructured text from the online tools to keywords from structured medical lexicons, such as UMLS (UMLS, 2012) and MeSH (MeSH, 2010). Of all the approaches to this mapping, the two most successful have been the Metamap program (Aronson, 2001) developed at the NLM, and Mgrep, the mapping tool of choice for the Open Biomedical Annotator (Jonquet et al., 2009).

These two programs take different approaches to the mapping process, and as such result in different sets of keywords when mapping the same source text. Previous research (Shah et al., 2009) has investigated comparing the two programs with respect to mapping the metadata associated with free, online databases, but this comparison did not explore the successes and failures of each program in any great detail, and the nature of metadata is very different from the archives of social media tools.

This paper is interested in comparing the results of using Metamap and Mgrep to map the archives of an unstructured medical mailing list to the MeSH medical lexicon. We first want to investigate general precision, to determine which program is more accurate with its mapping. We also want to delve deeper into the precision of the two programs, to determine if there is a relationship between mapping score and correctness, and we want to look at the overlap between the terms returned from the two programs.

The paper will proceed as follows: the background section will summarize the medical lexicon system, and the MeSH system in particular. It will explore some previous semantic mapping techniques, along with in depth explanations of how Metamap and Mgrep work. The methods section will outline the data preparation, the mapping process, and the analysis plan. The results section will summarize the analysis of the mappings by the two programs, and finally the discussion and conclusion sections will attempt to synthesize the analysis into a useful comparison of the two programs.

2 Background

In an evidence-based medical world, it is vital that knowledge be available to clinicians at the point of care. Unfortunately, the lack of organization, proper indexing, aging information sources and poor distribution have been shown to negatively affect a clinician's access to pertinent information (Covell et al., 1985; Timpka et al., 1989; Osheroff et al., 1991). The use of formal medical lexicons is a key step in improving clinician access to medical knowledge by providing a unified indexing of the existing medical knowledge.

The Unified Medical Language System (UMLS) is developed by the National Library of Medicine (NLM) to facilitate the computerization of medical knowledge, with the ultimate goal of allowing computer-systems to "understand" the meaning of biomedical and health text (UMLS, 2012). To this end they have created a number of tools, one of which is the "Metathesaurus", a formal lexicon that is the aggregate of over 150 different medical lexicons. The Metathesaurus includes a semantic network, assigning each term in the UMLS to one of the 135 generalized semantic types, which in turn have 54 relations between them. For a full listing of the UMLS Semantic Types, visit http://www.nlm.nih.gov/research/umls/META3_current_semantic_types.html.

The Medical Subject Headings (MeSH) lexicon is one of the subsets of the UMLS (MeSH, 2010). MeSH is the NLM's own controlled vocabulary, and is used to index the MEDLINE database. There are 26,142 terms in the 2011 edition of MeSH, arranged in a hierarchical fashion descending from 16 independent root nodes.

The UMLS and MeSH provide a valuable indexing resource to the medical profession, but clinicians need to be able to leverage these semantic languages in order to make full use of the formal indexing. Leroy and Chen (Leroy and Chen, 2001) developed a system that processes general medical queries and returns a set of medical keywords from the UMLS. Cimino et al (Cimino et al., 1993) designed a system that maps clinician queries to a set of generic queries based on UMLS keywords. Both of these systems take questions and map them to formal terms from a medical lexicon, which, though a first step, is different from mapping unstructured free text to a medical lexicon.

2.1 Semantic Mapping Techniques

The process of mapping free text to formal medical lexicons (and specifically to the UMLS) has long been an objective of the medical research community. The value of having formal representation of ideas combined with the challenge of performing the task manually has made research into automated approaches very valuable. This problem is often linked to MEDLINE, which is manually indexed by MeSH terms (MeSH, 2010), and thus provides an objective reason to connect text to UMLS terms. MicroMeSH (Lowe, 1987) was one of the first attempts to do this, by providing a simple system to expand search queries to MEDLINE and provide a tool where users could browse the MeSH tree around the terms they searched.

CHARTLINE (Miller et al., 1992) processed free text of medical records and connected them to relevant terms in the MeSH lexicon via a direct mapping. This process was improved by SAPHIRE (Hersh and Greenes, 1990), which explored the idea of processing free text and cleaning it by mapping terms to their synonyms. This was a valuable addition to the literature, as it normalized the process of mapping *women* to *woman*. This process was taken up by Nadkarni et al (Nadkarni et al., 2001) who used this synonym mapping along with a part of speech tagger to better identify the structure of the conversations and attempt to identify specific words and phrases in the text. PhraseX (Srinivasan et al., 2002) also used this kind of synonym parser to analyze the mapping of MEDLINE abstracts to the UMLS metathesaurus, in order to evaluate the contents of UMLS itself. Other, similar approaches include KnowledgeMap (Denny et al., 2003) and IndexFinder (Zou et al., 2003).

The current, gold standard is Metamap, though another product, called Mgrep (Shah et al., 2009) provides a very similar service. The creators of the Open Biomedical Annotator (OBA) (Jonquet et al., 2009) designed a system that leverages the results of any semantic mapping service (Metamap or Mgrep) and the ontology relations within the lexicon to produce a more complete semantic mapping. The OBA authors decided to make Mgrep their default mapping service, due largely to its vastly quicker processing times, but their approach would work with Metamap as well.

2.2 Metamap

Metamap uses a special natural language parser called SPECIALIST (Aronson, 2001) to find all the nouns and noun-phrases in a discussion thread, and maps them to one or more UMLS terms. Each mapped UMLS term is assigned a score that is a measure of how strongly the actual term mapped to the UMLS vocabulary. The score is a weighted average of four metrics measuring the strength of the matching, with an overall range in [0,1000], with higher scores indicating a better match. The formal equation for calculating the scores is:

$$\frac{1000 \times (Centrality + Variation + 2 \times Coverage + 2 \times Cohesiveness)}{6} \tag{1}$$

- Centrality: An indicator of whether the matched (source) term is the head of the phrase
- Variation: A measure of the distance between the matched term and the root word. For example, if the source word is eye and the match is to the term ocular, the distance is 2, as ocular is a synonym for eye
- Coverage and Cohesiveness: Measures of how well the source term and the UMLS term match each other: if the source and UMLS terms are both “pain” then the match is perfect, but if the source term ocular matches to the UMLS term Ocular Vision then the coverage and cohesiveness are less than perfect.

Metamap’s precision and recall in previous projects have varied depending on the format of the text being processed, from values as high as 0.897 and 0.930 respectively (Kahn and Rubin, 2009) to values as low as 0.56 and 0.72 (Chapman et al., 2004). The difference between the precision and recall values show that Metamap does a good job at returning pertinent MeSH terms, but also returns impertinent terms as well, i.e., its results are somewhat noisy. Projects that reported low recall and precision with Metamap acknowledged that many of the problems come from the inherently ambiguous nature of the text being processed: in processing medical residents’ voice recordings, it was noted that Metamap failed to recognize abbreviations, acronyms or complex phrases that omitted key terms (Chase et al., 2009).

For our purposes, the Metamap scoring system provides a baseline measure of how well the mapped UMLS term represents the original term in the PPML discussion thread. Table 1 contains some sample mappings to the MeSH lexicon and their scores.

Despite the inconsistencies in the terms returned by Metamap, it provides a valuable tool for mapping unstructured messages and conversations to a structured medical lexicon. The Knowledge Linkage project (Stewart and Abidi, 2012) uses these mappings to try and provide explicit knowledge links to the experiential knowledge being shared within the community.

2.3 Open Biomedical Annotator and MGrep

The Open Biomedical Annotator (Jonquet et al., 2009) was developed to automate the process of providing keywords to datasets that are available on the web. Their process was to take the metadata from the datasets, pass them through a semantic mapping engine (either Metamap or Mgrep) and then post-process their output using ontological relationships.

The authors of the Open Biomedical Annotator performed an experiment to compare MetaMap to Mgrep (Shah et al., 2009) in terms of accuracy and speed. They found that Mgrep performed slightly better in terms of precision and was much faster (1/5th of a second compared to 8 minutes). The authors concluded that, because they were looking for real-time implementation, Mgrep was was a better option for them, and thus The Open Biomedical Annotator was implemented using Mgrep.

The details of how Mgrep works are not completely clear, and publications on it have been limited to conference posters (Dai et al., 2008). The authors of the Open Biomedical Annotator claim that it "implements a novel radix-tree-based data structure that enables fast and efficient matching of text against a set of dictionary terms" (Jonquet et al., 2009). The scoring algorithm as well is not completely explained, though it performs a similar expansion scoring to Metamap, where partial matches and derived matches receive lower scores that perfect matches. Mgrep is not distributed itself, but is accessed via the OBA: performing a mapping with the OBA without using the ontological expansions results in a strictly Mgrep-based mapping. Table 1 contains some sample mappings from Mgrep.

The report stated that when music therapy is used, the babies required less pain medication. Does anyone know of any published reports of empirical research demonstrating the effect?					
Metamap Terms			**Mgrep Terms**		
Source	**MeSH Term**	**Score**	**Source**	**MeSH Term**	**Score**
music therapy	Music Therapy	1000	Music	Music	10
			therapy	therapy	10
the babies	Infant	966			
less pain medication	Pain	660	Pain	Pain	10
less pain medication	Pharmaceutical Preparations	827			
of any published reports	Publishing	694	Report	Report	16
			Research	Research	10
of empirical research	Empirical Research	1000	Empirical Research	Empirical Research	10

Table 1: Sample message and its associated MeSH mappings from both Metamap and Mgrep

2.4 Conclusion

It is clear that Metamap and Mgrep are the two most popular options for mapping medical free text to structured medical lexicons. Minimal research has been done in terms of comparisons, but more is needed, particularly within the mapping of social media data. Using MeSH as a target lexicon has the benefit of having many comparable projects, and the follow-up connection to MEDLINE and other sources that are indexed by MeSH is an additional reason to use it as a target lexicon.

3 Methods

The data for this project is the archives of the Pediatric Pain Mailing List (PPML) from January 2006 - December 2008. The data were originally extracted and processed for the Knowledge Linkages project (Stewart and Abidi, 2012) and the parsing and cleaning details are contained therein. For our purposes the content of the messages were extracted and cleaned to try and remove non-medical information (user signatures and reply-text being the major targets). An attempt was made to remove non-pertinent messages (such as conference announcements and job advertisements) as those types of messages do not contain the embedded medical knowledge that we are interested in. Once the data was cleaned and prepared it was mapped with both Metamap and the Open Biomedical Annotator (OBA), producing a set of terms and scores for each message from each program.

3.1 Mapping

In a paper by Abidi (Abidi et al., 2005) they outlined semantic filters they applied when using Metamap in mapping the content of clinical practice guidelines to formal medical terms. Of the 135 semantic types in the UMLS certain types, such as Amphibian or Professional Society, were not deemed pertinent to the subject, and were filtered out. 108 of the semantic types were used, while 27 were filtered out. The semantic types filtered out were: Amphibian, Animal, Bird, Class, Family Group, Fish, Functional Concept, Geographic Area, Group, Idea or Concept, Intellectual Product, Language, Mammal, Occupation or Disciple, Organization, Physical Object, Plant, Population Group, Professional Society, Professional or Organizational Group, Qualitative Concept, Quantitative Concept, Regulation or Law, Reptile, Research Device, Self-help or Relief Organization, Spatial Concept, Temporal Concept and Vertebrate.

The mapping was done using Metamap09. Though newer versions of Metamap have been made available the decision was made to use the same mappings that were done in the original project (Stewart and Abidi, 2012). Changes between versions of Metamap are minimal, so a change to the new version of the program is not expected to drastically affect the results.

For Mgrep, the mapping was done using the OBA REST services, available at http://bioportal.bioontology.org/annotator. The OBA has the same semantic type filters as Metamap, and the same filtering set was used. None of the OBA expansion options were used, resulting in the OBA returning a strictly Mgrep-mapped set.

In order to make the scores comparable between the programs, the Metamap scores were divided by 100, putting them on the same [0, 10] range as the Mgrep scores. For each program, the terms within a specific message were aggregated. This means that, though the range for an individual mapping score is [0,10], the scores can in reality go from [0,∞], as there could be multiple mappings of the same term in a message. For the mappings reviewed, the maximum score returned was 128.26 for Metamap and 190 for Mgrep.

Once the mappings were created they needed to be checked. The messages and their mappings were reviewed by a medical expert. For each message the content of the message was first evaluated to determine if it was medically oriented, completing the filtering process that was somewhat handled in the data cleaning process. After that each MeSH term mapped to the message was reviewed and determined to be relevant to the conversation or not. The process was continued until 200 medically relevant messages had been found, with 127 messages being deemed not medically relevant.

3.2 Analysis

The analysis will begin with a simple investigation of the precision of both programs. Since both programs report scores for each mapping, an investigation of the relationship between score and correctness will also be investigated, to determine both the value of the scores being returned, and whether the scores could be used to improve the mapping process. We also want to compare the mappings between Mgrep and Metamap to study the overlap. The natural partner when studying precision is recall, but while precision, the proportion of returned terms that are correct, is relatively simple to calculate, recall, the number of correct terms that were found, is not nearly as simple to find, as this requires the correct terms for each of the messages to be pre-specified, which was not a feasible task for this project. Relative recall (Clarke and Willett, 1997) is often used to compare search strategies in which there is no annotated database to calculate recall from, but relative recall tends to favour system that return more results, and Metamap returned many more terms, and thus must have a higher relative recall. We will instead look at the overlap between the two programs and its relationship to precision.

4 Analysis

Table 2 presents some summary statistics for both Mgrep and Metamap. As we can see in the table, Mgrep had significantly higher precision, with a p-value < 0.0001.

Program	# terms	# correct	Precision	difference	p-value
Metamap	2381	1384	58.12%		
Mgrep	1350	1027	76.07%	17.95% [14.9%,21.0%]	< 0.0001

Table 2: Summary of the mapping process for both programs. The p-value is calculated using a 2-sample z-test with a continuity correction.

4.1 Scores and Correctness

Though Mgrep has a higher general precision than Metamap, the relationship between score and correctness reveals that Metamap's precision may be better than it appears.

Figure 1 presents boxplots for both programs, comparing the scores for both programs between incorrect and correct mappings.

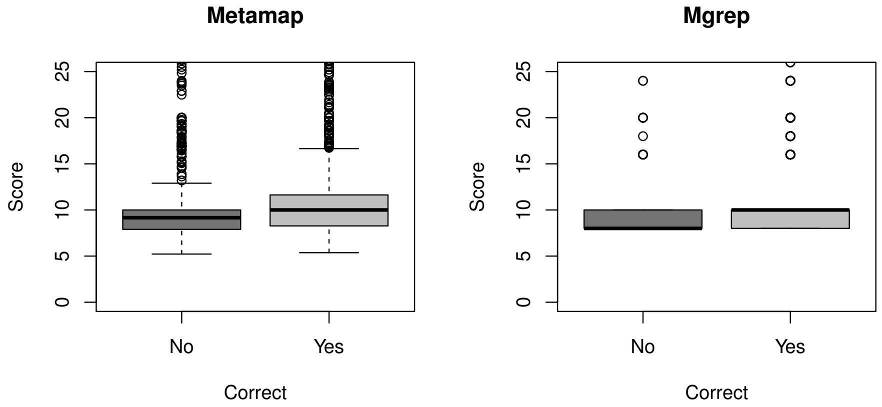


Fig. 1: Boxplots for comparing scores to correctness for both programs. Note that the plots are truncated to the [0,25] range for clarity.

For both programs it appears that there is a significant relationship between score and correctness, though the difference is more pronounced for the Metamap scores, as that program returns a wider range of scores. Infact, for individual terms Mgrep does not seem to return scores other than 8 or 10, with higher scores resulting from multiple mappings within the same message. Table 3 presents the comparison of correctness to score, and finds that, for both programs the correct terms have significantly higher scores.

		n	**mean**	**Quantiles** [5%,25%,50%,75%,95%]	**Mean diff.**	**p-value**
Metamap	*Correct*	1384	12.40	[6.38, 8.27, 10.00, 11.63, 28.27]		
	Incorrect	997	9.82	[5.94, 7.89, 9.16, 10.00, 19.01]	2.57	< 0.0001
Mgrep	*Correct*	1027	13.68	[8, 8, 10, 10, 30]		
	Incorrect	323	10.13	[8, 8, 8, 10, 17.8]	3.55	< 0.0001

Table 3: Comparing scores to correctness for both programs. The p-values are calculated using a Wilcoxon Rank-Sum test to account for the extreme skewness of the data.

The relationship between scores and correctness can be investigated further by looking at 10% quantiles of the data. Tables 4 and 5 report the correctness stratified by 10% quantiles of the scores. The quantiles of the Metamap scores are much more spread out, which is to be expected as their scoring algorithm is more complex, resulting in a wider range of values. What is interesting, looking at the table, is that there seems to be a significant jump in precision for both programs for terms that score 10 points or higher. Table 6 looks at the relationship between correctness and score dichotomized to above/below 10 points.

Quantile	[5.22,6.6)	[6.6,7.55)	[7.55,8.61)	[8.61,8.75)	[8.75,9.28)	[9.28,10)	[10,18.6)	[18.6,128)
Correct	129	77	135	94	109	149	247	57
Incorrect	94	175	104	143	56	56	554	201
n	223	252	239	237	165	205	801	258
Precision	0.42	0.69	0.44	0.60	0.34	0.27	0.69	0.78

Table 4: Correctness by 10% quantiles of scores for Metamap. Note that quantiles that were the same were collapsed together, thus the quantile [10, 18.6) has 801 observations in it, which represents 3 quantiles of data.

Quantiles	[8,10)	[10,16)	[16,20)	[20,190)
Correct	162	126	19	16
Incorrect	328	445	69	184
n	490	571	88	200
Precision	0.67	0.78	0.78	0.92

Table 5: Correctness by 10% quantiles of scores for Mgrep. Because of the lack of range of Mgrep scores many of the quantiles were similar, and were thus collapsed into 4 groups from 10.

Metamap's precision has jumped from 58% to 71%, while Mgrep's has jumped from 76% to 81%. Though Mgrep's precision amongst only those terms that score ≥ 10 is still significantly higher (10% difference, 95% CI: [6.1%, 13.9%], p-value < 0.0001), Metamap improved it's precision by 13%, whereas Mgrep only improved by 5%. It is clear that there is a significant relationship between score and correctness.

	Metamap Score		
	< 10	≥ 10	*Total*
Correct	628	756	1384
Incorrect	693	304	997
Total	1321	1060	2381
Precision	47.5%	71.3%	

m

	Mgrep Score		
	< 10	≥ 10	*Total*
Correct	328	699	1027
Incorrect	162	161	323
Total	490	860	1350
Precision	66.9%	81.2%	

Table 6: Looking at the relationship between score ≥ 10 and correctness for both programs.

4.2 Overlapping Terms

The overlap between the terms returned by Metamap and Mgrep presents an opportunity to try and evaluate the recall of the two programs. Though formal recall cannot be calculated, and relative recall is not valuable when one program returns so many more terms, studying what terms one program returned that another did not, and investigating what terms are missing, presents a valuable comparison of the two programs. Table 7 presents the overlap of the two programs with respect to correctness, and Figure 2 provides a visual representation of the difference.

Program	*Incorrect*	*Correct*	*Precision*	*Total*
Metamap Only	800	621	0.437	1421
Mgrep Only	126	264	0.677	390
Both Programs	207	782	0.791	989

Table 7: Comparing the overlap of the two programs. The precision reported is the number of terms for that row that are correct, i.e., it is the *Correct* column divided by the *Total* column.

The overlap of the two programs presents some interesting results. Of the 1350 terms returned by Mgrep, 989 were also returned by Metamap, resulting in an overlap of 73%. With 2381 terms returned, 41% of the terms returned by Metamap were also covered by Mgrep. Put in another way, if one were to only use Metamap, there would have been 264 correct mappings that were missed, while if one were to only use Mgrep there would be 621 correct mappings missed.

As demonstrated in Figure 2, the terms where the programs overlapped were more likely to be correct, with an overlap precision of 79.1%. This also leads to both programs having lower precision on the terms that only they returned than their overall average precision.

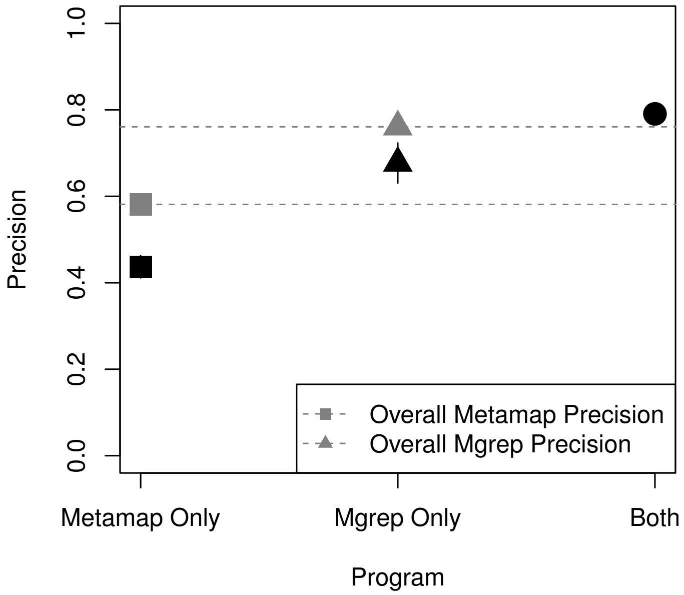


Fig. 2: Comparing the overlap of the two programs to their precision.

5 Discussion

Based strictly on precision, Mgrep outperforms Metamap. A difference of nearly 18% confirms the findings of (Shah et al., 2009) in their original investigation of the two programs. There is much more depth to the comparison, however, which reveals the potential utility of Metamap in certain situations.

Though both programs provide mapping scores, Metamap's seem more useful, providing both a wider range of scores and a larger difference in precision between the low and high scoring terms. One of the challenges of this comparison is a lack of details on how the Mgrep scoring algorithm works, but, though the authors claim a range of [0,10], in reality only 8's and 10's were returned (with higher scores all being aggregates of those two numbers).

Of particular interest is the poor performance of terms returned by Metamap that have scores just below perfect: Looking back at Table 4, the fifth decile, [8.75, 9.28), has a precision of only 34%. Looking into the mappings in this quantile, we see mappings that are based on variations in the root word, along with words that are based on a less than perfect coverage. The mappings in this group are inaccurate because they are taking a source term like "replacing" and mapping it to the MeSH term "Replantation", which is rarely going to be the correct mapping. In an attempt to dig deeper into the potential variations on source terms, Metamap seems to be hurting its overall precision.

When mappings are restricted to only perfect matches (or less than perfect matches that occur multiple times), the precision of both programs increases, but the increase is more dramatic for Metamap (see Table 6). Previous studies that have investigated Metamap could improve their results by putting more effort into leveraging the Metamap scores.

This does not mean that terms that score less than perfect should necessarily be dropped, however, as there is a more to the evaluation of the two programs than precision. Looking back at table 6, removing all mappings with scores < 10 would drop 693 correct Metamap mappings and 162 correct Mgrep mappings. If the objective of the mapping process is strictly about precision then this may be a logical step, but if the objective is to try and find suggested terms to provide to the users, then there is little harm in providing incorrect suggestions, especially if it leads to more pertinent terms being provided as well. Looking at the overlap of the two programs, though Mgrep had a higher precision, it missed 621 terms that Metamap provided, terms which may have been beneficial to the user. Likewise, there are 264 terms missed by Metamap that were returned by Mgrep, which could also have been helpful.

If the objective of the mapping process is strictly to be as precise as possible, then using Mgrep and restricting the mapping solely to terms that score 10 points will result in the most accurate mapping. If you are developing a suggestion engine, however, or if your system can leverage the mappings scores, as our Knowledge Linkage program did (Stewart and Abidi, 2012), then perhaps the larger set returned by Metamap, combined with the superior scoring function, may be more useful to your project.

Though it was not studied formally in this project, we did find that Mgrep was vastly faster than Metamap, even when used over the internet through their REST services. This confirms the findings of (Shah et al., 2009), and if you are trying to develop a real-time system then Metamap may be too slow for your application.

6 Conclusion

There is an obvious need for indexing engines that can process free text and match them to formal medical lexicons. Though this project focused on MeSH, there are obvious expansions to any component of the UMLS, and mappings to ICD and SNOMED can provide valuable resources to those working in health information technology.

The mapping of social media archives to MeSH is a challenging objective. A precision of 58% by Metamap is at the low end of the range of precisions reported by other papers that studied program (Chapman et al., 2004; Chase et al., 2009), and the challenges of mapping abbreviations, acronyms and complex phrases from medical charts continue to be a problem for the mapping of social media data. This does not mean that the mapping process cannot be used, but when leveraging the terms provided by these programs the potential for incorrect mappings must be taken into account.

This project had some shortcomings. A double review of the mappings rather than a single review would have provided more confidence in the “correctness” of the mappings. The Metamap program used was the 2009 edition, as those were the mappings that were produced for the Knowledge Linkage project (Stewart and Abidi, 2012), and there have been multiple releases since then. Re-running the analysis with the new program would probably not change the precision of Metamap significantly, but it would certainly change some of the mappings. We believe that the general structure of the analysis would remain the same, however a comparison of the old and new versions should be investigated. More details of how Mgrep works need to be made available, especially with respect to the scoring algorithm. As well, the aggregation of multiple mappings needs to be broken down, which could be used to expand the results in section 4.1. Correct/Incorrect may not be the best way to classify mappings: providing the term "Pain" in a discussion of needle stick injuries is not incorrect, but it is not as useful as the MeSH term "Needle Stick". Re-evaluating each mapping on a 5-point Likert Scale may provide more valuable insights.

Developing a way to measure some form of recall would improve the analysis: studying the crossover between the two programs is helpful, but being able to identify and study what was missed is a valuable component of the comparison of the two programs. Each message could be reviewed, and the potential MeSH terms that are not present could be recorded, providing some insight into terms that were not mapped. This analysis will be done in future work.

Moving forward, the programs are best measured not by evaluating their correctness in terms returned, but by their utility embedded in other programs. Re-implementing the Knowledge Linkage project with Mgrep and re-running the analysis from that project (Stewart and Abidi, 2012) would be a stronger way to measure whether Mgrep is more or less useful in mapping free text to medical lexicons. A larger review set would also allow a more indepth analysis of the correctness as a function of position in the MeSH tree, both in terms of source root and depth from the top.

Bibliography

Abidi, S. (2006). *Healthcare Knowledge Sharing: Purpose, Practices, and Prospects*, chapter 6, pages 65–86.

Abidi, S., Kershaw, M., and Milios, E. (2005). Augmenting gem-encoded clinical practice guidelines with relevant best evidence autonomously retrieved from medline. *Health Informatics Journal*, 11(2):95–110.

Aronson, A. R. (2001). Effective mapping of biomedical text to the umls metathesaurus: The metamap program. *Proceedings of the AMIA Symposium*.

Chapman, W. W., Fiszman, M., Dowling, J. N., Chapman, B. E., and Rindflesch, T. C. (2004). Identifying respiratory findings in emergency department reports for biosurveillance using metamap. *MEDINFO*.

Chase, H. S., Kaufman, D. R., Johnson, S. B., and Mendonca, E. A. (2009). Voice capture of medical residents' clinical information needs during an inpatient rotation. *Journal of the American Medical Informatics Association*, 16:387–394.

Cimino, J., Aguirre, A., Johnson, S., and Peng, P. (1993). Generic queries for meeting clinical information needs. *Bulletin of the Medical Library Association*, 81(2):195–206.

Clarke, S. J. and Willett, P. (1997). Estimating the recall performance of web search engines. In *Aslib Proceedings*.

Covell, D., Uman, G., and Manning, P. (1985). Information needs in the office practice: are they being met? *Annals of Internal Medicine*, 103(4):596–599.

Dai, M., Shah, N., Xuan, W., Musen, M., Watson, S., Athey, B., and Meng, F. (2008). An efficient solution for mapping free text to ontology terms. *AMIA Summit on Translational Bioinformatics,*.

Denny, J. C., Smithers, J. D., Miller, R. A., and Spickard, A. (2003). understanding medical school curriculum content using knowledgemap. *JAMIA*, 10:351–362.

Hersh, H. and Greenes, R. (1990). Saphire - an information retrieval system featuring concept matching, automatic indexing, probabilistic retrieval, and hierarchical relationships. *Comput Biomed Res*, 23:410–425.

Jonquet, C., Shah, N. H., and Musen, M. A. (2009). The open biomedical annotator. *Summit of Translational Bioinformatics*, pages 56–60.

Kahn, C. E. J. and Rubin, D. L. (2009). Automated semantic indexing of figure captions to improve radiology image retrieval. *Journal of the American Medical Informatics Association*, 16:280–286.

Leroy, G. and Chen, H. (2001). Meeting medical terminology needs–the ontology-enhanced medical concept mapper. *IEEE Transactions on Information Technology in Biomedicine*, 5(4):261–270.

Lowe, H. (1987). Micromesh: a microcomputer system for searching and exploring the national library medicines medical subject headings (mesh) vocabulary. *Proc Annu Symp Comput Appl Med Care*, pages 717–20.

MeSH (2010). Medical subject headings. http://www.nlm.nih.gov/mesh/.

Miller, R. A., Gieszczykiewicz, F. M., Vries, J. K., and Cooper, G. F. (1992). Chartline: Providing bibliographic references relevant to patient charts using the umls metathesaurus knowledge sources. *Proc Annual Symposium of Comput Appl Med Care*, pages 86–90.

Nadkarni, P., Chen, R., and Brandt, C. (2001). Umls concept indexing for production databases: a feasibility study. *JAMIA*, 8:80–91.

Osheroff, J., Forsythe, D., Buchanan, B., Bankowitz, R., Blumenfeld, B., and Miller, R. (1991). Physicians' information needs: analysis of questions posed during clinical teaching. *Annals of Internal Medicine*, 114(7):576–581.

Shah, N. H., Bhatia, N., Jonquet, C., Rubin, D., Chiang, A. P., and Musen, M. A. (2009). Comparison of concept recognizers for building the open biomedical annotator. *BMC Bioinformatics*, 10 (suppl 9):S14.

Srinivasan, S., Rindflesch, T. C., Hole, W. T., Aronson, A. R., and Mork, J. G. (2002). Finding umls metathesaurus concepts in medline. *Proc AMIA Symp*, pages 727–731.

Stewart, S. A. and Abidi, S. S. R. (2012). An infobutton for web 2.0 clinical discussions: The knowledge linkage framework. *IEEE Transactions on Information Technology in Biomedicine*, 16(1):129–135.

Timpka, T., Ekstrom, M., and Bjurulf, P. (1989). Information needs and information seeking behavior in primary health care. *Scandanavian Journal of Primary Health Care*, 7(2):105–109.

UMLS (2012). Unified medical language system fact sheet. Web. http://www.nlm.nih.gov/pubs/factsheets/umls.html.

Wenger, E. (2004). Knowledge management as a doughnut: Shaping your knowledge strategy through communities of practice. *Ivey Business Journal*, pages 1–8.

Zou, Q., Chu, W. W., Morioka, C., Leazer, G. H., and Kangarloo, H. (2003). Indexfinder: A method of extracting key concepts from clinical texts for indexing. *AMIA Annu Symp Proc*, pages 763–767.

Exploring the Similarity between Social Knowledge Sources and Twitter for Cross-domain Topic Classification

Andrea Varga, Amparo Elizabeth Cano and Fabio Ciravegna[1]

OAK Group,
Dept. of Computer Science,
The University of Sheffield,
United Kingdom
firstinitial.lastname@dcs.shef.ac.uk

Abstract. The rapid rate of information propagation on social streams has proven to be an up-to-date channel of communication, which can reveal events happening in the world. However, identifying the topicality of a short messages (e.g. tweets) distributed on these streams poses new challenges in the development of accurate classification algorithms. In order to alleviate this problem we study for the first time a transfer learning setting aiming to make use of two frequently updated social knowledge source (KS) (DBpedia and Freebase) for detecting topics in tweets. In this paper we investigate the similarity (and dissimilarity) between these KS and Twitter at the lexical and conceptual(entity) level. We also evaluate the contribution of these types of features and propose various statistical measures for determining the topics which are highly similar or different in KS and tweets. Our findings can be of potential use to machine learning or domain adaptation algorithms aiming to use named entities for topic classification of tweets. These results can also be valuable in the identification of representative sets of annotated articles from the KS, which can help in building accurate topic classifiers for tweets.

Keywords: social knowledge sources, transfer learning, named entities, data analysis

1 Introduction

Micropost platforms such as Twitter serve as a real-time channel of information regarding events happening around the world. Compared to traditional news sources, microposts communicate more rapidly up-to-date information on a large number of topics. Identifying these topics in real-time could aid in different scenarios including i.e., emergency response, and terrorist attacks.

However, microposts mining poses several challenges since some of the characteristics of a tweet include: i) *the use of non-standard English*; ii) *the restricted size of a post (limited to 140 characters)*; iii) *the frequent misspellings and use of jargon*; and iv) *the frequent use of abbreviations*.

The dynamic changes in both vocabulary and style pose additional challenges for supervised classification algorithms, since the collection of annotated data becomes particularly difficult. However, frequently updated social knowledge sources(KS), such as DBpedia and Freebase, present an abundant source of structured data which could potentially aid in streamed topic detection. Similar to Twitter, these sources exhibit the following characteristics: i) *they are constantly edited by web users*; ii) *they are social and built on a collaborative manner*; iii) *they cover a large number of topics*; and iv) *they provide plentiful amount of annotated data*.

In this work we present *for the first time* a comparative study which analyses the similarity between Twitter and two frequently updated KS including *DBPedia* and *Freebase*. This comparative study includes the analysis of various cross-domain(CD) topic classifiers built on these KSs considering different lexical and conceptual features derived from named entities. Our intuition for the conceptual features is that the mention of certain entity types could be a good indicator for a specific topic. For e.g. a tweet containing the entity "Obama" is more likely to be a trigger for the topics "Politics" and "War&Conflict" than for the topic "Entertainment". Similarly, "Lady Gaga" is more likely to appear in tweet messages about the topics "Entertainment" or "Music", than about the topic "Sports".

In addition, we propose different statistical measures for quantifying the similarity and differences between these KS and tweet messages. The main research questions we investigate are the following: *i*) *Do KSs reflect the lexical changes in Twitter?*; *ii*) *Which features make the KSs look more similar to Twitter?*; *iii*) *How similar or dissimilar are KS to Twitter*; and *iv*) *Which similarity measure does better quantify the lexical changes between KS and Twitter?*

The main contributions of this paper are as follows: *i*) *we present a methodology for building CD topic classifiers for tweets making use of KSs*; and *ii*) *we present a comparative analysis exploring the similarity between KSs and Twitter at the level of words and named entities for CD topic classification*;

In the remaining of the paper we briefly describe the DBpedia and Freebase KS, we then present the state-of-the-art approaches in topic classification of Tweets, then we describe the main methodology and present the results obtained.

2 Social Knowledge Sources: an overview of DBpedia and Freebase

In this section we briefly review the main features of the DBpedia and Freebase KSs, highlighting the differences and similarities between them.

DBpedia[1] is a structured knowledge base derived from Wikipedia[2], the largest collaboratively maintained encyclopaedia. The latest released, DBpedia 3.7, classifies 1.83 million resources into 740,000 Wikipedia categories and 18,100,000 YAGO2 categories. For a given Wikipedia article DBpedia provides the following information [4]: *i*) the *title* of the Wikipedia article; *ii*) the *abstract* of the article corresponding to the first few

[1] http://dbpedia.org

[2] http://wikipedia.org

paragraphs containing up to 500 words; *iii*) the Wikipedia *categories* (topics) assigned to the article; *iv*) various links such as the *external links* pointing to external Web resources, *redirects* pointing to other articles about synonymous terms, *pagelinks* describing all the links in the article, *inter-language links* pointing to the translations of the article into multiple languages; *v*) *disambiguation pages* explaining different meaning of homonyms about a given term; *vi*) *images* depicting the resources from the article; *vii*) *homepage* or *website* information for an entity such as organisation or company; and *viii*) *geo-coordinates* of a particular resource of the article.

Similarly, Freebase[3] is a huge online knowledge base which users can edit in a similar manner as Wikipedia. The latest version of Freebase [4] comprises of 85 domains, more than 20 million entities and more than 10 thousand relations across a large number of these domains. In contrast to DBpedia however, in Freebase the source of articles include Wikipedia as well as other sources such as MusicBrainz, WordNet, OurAirports, etc [5]. The classification of articles in Freebase is also slightly different; for a given Freebase article: *i*) a *domain* denote the topic of the article; *ii*) a *type* define a particular kind of entity such as person or location (for e.g. “Lady Gaga” is a Person); and *iii*) *properties* describe an entity (for e.g. “Lady Gaga” has a “place of birth”). Another notable difference between the two knowledge source is the level of deepness in the hierarchy for a particular category or topic.

3 Related Work

DBpedia and Freebase KSs have been important knowledge sources in many classification tasks such as topic detection and semantic linking of Twitter messages. These approaches mostly employ traditional machine learning algorithms building a classifier on Twitter dataset and deriving useful features from KSs.

To date, to the best of our knowledge, no analysis has been done in exploiting these KSs for cross-domain (CD) topic classification of tweets and also in measuring the similarity between these KSs and Twitter. In the following section we thus provide a summary of the related work using these KSs for Twitter on topic detection and semantic linking.

Related Work on using DBpedia for Topic Classification of Tweets Ferragina et al. [7] propose the TAGME system, which enriches a short text with Wikipedia links by pruning n-grams unrelated to the input text. Milne et al. [11] propose an automatic cross-reference of Wikipedia documents and Wikipedia links by means of machine learning classifiers. This method has been shown to not perform well when applied to tweets [10]. Munoz et al [1] also address the problem of assigning labels to microposts, in order to identify what a micropost is about. In their approach they assign DBpedia resources to post by means of a lexicon-based similarity relatedness metric.

Meij et al [10], also assign resources to microposts. In their approach they make use of Wikipedia as a knowledge source, and consider a Wikipedia article as a *concept*,

[3] http://www.freebase.com/

[4] `http://download.freebase.com/datadumps/2012-07-19/`

[5] `http://wiki.freebase.com/wiki/Data_sources`

their task then is to assign relevant Wikipedia article links to a tweet. They propose a machine learning approach which makes use of Wikipedia n-gram and Wikipedia link-based features. Our approach differs from theirs in two main points: 1) rather than considering a Wikipedia article or DBPedia resource link as a concept, we consider a whole DBpedia category as a concept; 2) our study analyses the use of DBpedia as an annotated source dataset, which can be used to increase the performance of machine learning classifiers for assigning a topic label to a tweet.

Mendes et al. [12] propose the Topical Social Sensor, which is a system that allows users to subscribe to hashtags and DBpedia concepts in order to receive updates regarding these topics. They link a tweet with the DBpedia concepts derived from the entities contained in it. This system is designed for detecting a hype on a topic defined a priori. In our work rather than relating a tweet with the DBpedia concepts derived from named entities, we propose the use of DBpedia articles to model a category, and perform an use this articles as source dataset for training a topic classifier to assign a topic label to a tweet.

Related Work on using Freebase for Topic Classification of Tweets Kasiviswanathan et al[9] propose a detection-clustering based approach for streamed topic detection they make use of entities and their types gathered from Freebase. In this paper, rather than proposing a new approach for topic detection we compare the performance of two classifiers; one based on DBpedia and the other on Freebase for detecting topics of tweets.

4 Methodology

This section describes three different steps required for the analysis presented in this paper. The first step, described in Section 4.1, consists on the compilation of datasets from KSs; the second step, described in Section 4.2, consists on the use of these datasets for the development of CD topic classifiers; and the third step consists on the introduction of similarity metrics that can characterise distributional changes between datasets.

4.1 Collecting Data from KS

In this section we refer to our datasets, which will be further described in Section 5. The Twitter dataset consists of a collection of tweets, which were annotated with 17 different topics using the OpenCalais services. In order to compile a set of articles relevant to each of these 17 topics, from both DBpedia and Freebase KSs, we performed two steps. In the case of DBpedia, for a given topic, we SPARQL[6] queried for all resources whose categories and subcategories are similar to the topic. For the returned resources we only kept the first 500 characters from the resources' abstracts. In the case of Freebase, we downloaded the articles using the Freebase Text Service API[7]. Given a topic, we collected all the articles whose domain matched the topic [8]. In addition, for some of the

[6] http://www.w3.org/TR/rdf-sparql-query/

[7] `http://wiki.freebase.com/wiki/Text_Service`

[8] The collection of domains are enumerated at `http://www.freebase.com/schema`

topics (e.g. Disaster or War), which were not defined as domains in Freebase, we looked at all the articles containing these topics in their title. While this service also allows to download the full content of an article, similarly to DBpedia, we only considered the first paragraph up to 500 characters.

The following Subsection 4.2, describes how the DBpedia and Freebase datasets are used to built three different CD classifiers for detecting topics in tweets.

4.2 Building Cross-Domain(CD) Topic Classifier of Tweets

We formally describe each dataset D as a tuple $(X, F, P(X))$ composed of a set of *instances* X, a set of *features* F and a *marginal probability distribution* $P(X)$. Each instance $x \in X$ is represented by a vector of features $x = (f_1, .., f_m), f_i \in F$. The possible *topics* $y = \{cat_{Y_1}, \ldots, cat_{Y_d}\}$ for an instance x can take values from $Y \in \{cat_1, \ldots, cat_k\}$. The goal of the classification then is to learn a model $h : X \rightarrow Y$ from a set of annotated *training data* $L = \{(x_1, y_1), \ldots, (x_n, y_n) | x_i \in X, y_i \in Y\}$, that induces a non-injective map between X and Y such that multiple class labels can be assigned to the same instance - e.g. $h(x_1) = \{cat_1, cat_2\}$.

A CD scenario consists of a *source dataset* $D_S = (F_S, X_S, P(X_S))$ –on which the classifier is built– and a *test dataset* $D_T = (F_T, X_T, P(X_T))$ –on which the classifier is evaluated–. As illustrated in Figure 1, in this paper we consider three cases for the source dataset. The first two cases aim to investigate the usefulness of DBpedia and Freebase KSs independently, and the third case combines the contribution of both KSs. Thus, the CD scenarios studied in the paper are described as follows: **Scenario I (Sc.DB)** consisting of *sole DBpedia articles*; **Scenario II (Sc.FB)** consisting of *sole Freebase articles*; and **Scenario III (Sc.DB-FB)** consisting of a *joint set of DBpedia and Freebase articles*. The test dataset in each case is the Twitter dataset.

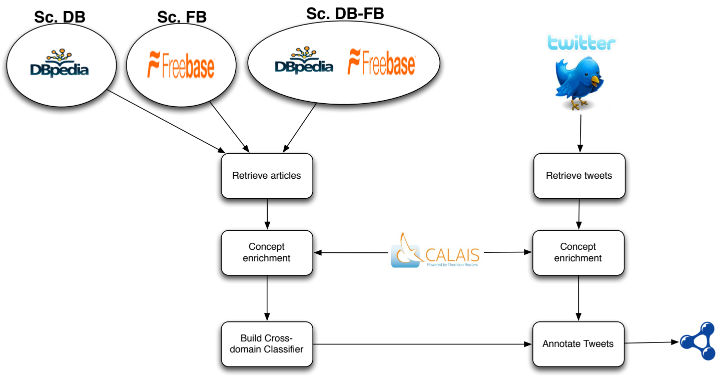


Fig. 1. The Sc.DB, Sc.FB and Sc.DB-FB CD scenarios using concept enrichment.

We used as baseline classifier an SVM classifier with linear kernel, which has been found to perform best for transfer learning [6]. We also took the commonly used one-vs-all approach to decompose our multi-label problem into multiple independent binary classification problems.

Feature Extraction The performance of the machine learning algorithm rely on the feature representation employed. We propose two different feature sets for the examples in both train and test datasets:

- *a bag-of-word*(***BoW***) representation: This representation captures our natural intuition to utilise what we know about a particular topic, so that the features which are most indicative of a topic can be detected and the appropriate label(s) assigned. This feature consists of a collection of words weighted by TF-IDF (term frequency-inverse document frequency) in order to capture the relative importance of each word.
- *a bag-of-entities*(***BoE***) feature representation. The second set of features makes use of named entities. These entities were extracted by querying OpenCalais API[9] for entity extraction on each instance belonging to the Dbpedia, Freebase and Twitter datasets as presented in Figure 1. We then used the dereferenceable URI and concepts returned by the API as features for the classifier. Our intuition is that entities can be characteristic of a topic, serving as trigger words for this topic; reducing in this way the lexical differences between the source and target datasets.

4.3 Measuring Distributional Changes Between KS and Twitter

In addition to building the CD classifiers, we investigated various measures for quantifying the similarity between KSs and Twitter. When building a machine learning classifier, it is expected that the closer the train dataset to the test dataset the better the performance of the classifier [13]. Therefore, these similarity metrics can be potentially useful in predicting the adequacy of the data collected from a KS in detecting topics in tweets.

To measure the similarity between the distributions of the presented datasets, let $\overrightarrow{d}$ represent a vector consisting of all the features occurring on a dataset. Then, $\overrightarrow{d_s}$ denotes such a vector for the train dataset and $\overrightarrow{d_t}$ for the test dataset. In light with the feature set employed, the $\overrightarrow{d_s}$ and $\overrightarrow{d_t}$ contain the TF-IDF weight for either the **BoW** or **BoE** feature sets. Then the proposed statistical measures are:

- the *chi-squared (χ^2) test*: The χ^2 test measures the independence between the feature sets (F_S and F_T) and the train and test datasets. Given the $\overrightarrow{d_s}$ and $\overrightarrow{d_t}$ vectors , the χ^2 test can be computed as

$$\chi^2 = \sum \frac{(O - E)^2}{E}$$

  , where O is the observed value for a feature, while E is the expected value calculated on the basis of the joint corpus.

[9] www.opencalais.com/

- the *Kullback-Leibler symmetric distance* (KL): Originally introduced in [3], the symmetric KL divergence metric measures how different the $\overrightarrow{d_s}$ and $\overrightarrow{d_t}$ vectors are on the joint set of features $F_S \cup F_T$:

$$KL(\overrightarrow{d_s}||\overrightarrow{d_t}) = \sum_{f \in F_S \cup F_T} (\overrightarrow{d_s}(f) - \overrightarrow{d_t}(f)) \log \frac{\overrightarrow{d_s}(f)}{\overrightarrow{d_t}(f)}$$

- *cosine similarity* measure: The cosine similarity represents the angle that separates the train and test vectors $\overrightarrow{d_s}$ and $\overrightarrow{d_t}$:

$$cosine(\overrightarrow{d_s}, \overrightarrow{d_t}) = \frac{\sum_{k=1}^{F_S \cup F_T} (\overrightarrow{d_s}(f_{S_k}) \times \overrightarrow{d_s}(f_{T_k}))}{\sum_{k=1}^{F_S \cup F_T} (\overrightarrow{d_s}(f_{S_k}))^2 \times \sum_{k=1}^{F_S \cup F_T} (\overrightarrow{d_t}(f_{T_k}))^2}$$

We also note that some of these proposed functions measure actual similarity ($cosine$), while others measure distance KL, χ^2.

5 Dataset and Data Pre-Processing

The Twitter dataset consists of tweets posted between October 2010 and January 2011, and was originally collected by [2],[10] comprising more than 2 million Tweets posted by more than 1619 users. We further annotated this data set with topics returned by the OpenCalais service, which label each tweet with one or more topics (from a collection of 17 topics). For our analysis we randomly selected one thousand tweets for each topic, excluding re-tweets, resulting in a collection of 12,412 Tweets. Some of these categories are presented in Table 1. Similarly from DBpedia and Freebase we randomly selected one thousand articles for each topic, comprising of 9,465 articles from DBpedia and 16,915 articles from Freebase.

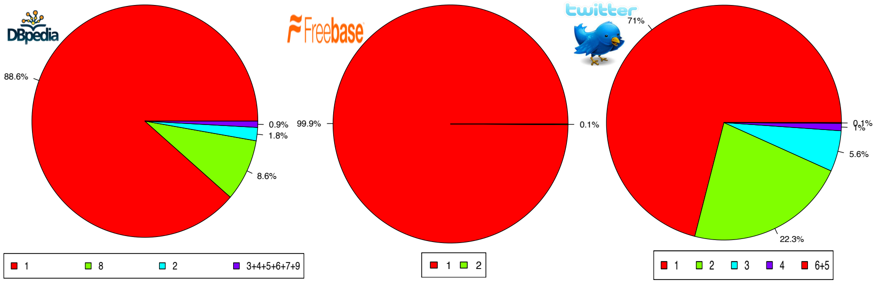


Fig. 2. The multi-label distribution in DBpedia, Freebase and Twitter datasets. The numbers in the legend indicate the number of topics assigned to an example, varying from 1 to 9 topics.

[10] Available at http://wis.ewi.tudelft.nl/umap2011/

In line with previous approaches ([2]), for the datasets we removed all the stopwords and we converted all words into lower case; after which a Lovins stemmer was applied. In addition, in order to reduce the vocabulary differences between the KS datasets and Twitter, all hashtags, mentions and URL links, which are particular to the Twitter dataset, were removed. The feature space was also reduced to the top-1000 words weighted by TF-IDF for each category.

Figure 2 shows the distribution of the examples belonging to multiple topics in each dataset. The Twitter dataset contain some tweets annotated with up to six categories, with the majority of them being annotated with only one topic. In the case of the Freebase dataset, due to the nearly flat hierarchical structure of the domains, the majority of the articles belong to a single category. In the case of the DBpedia dataset the majority of the articles belong to a single category, and less than 1% of the articles are annotated with 3,4,5,6,7 or 9 topics. The size of the vocabulary for each category and dataset is presented in Figure 3. This distribution presents a variation in the vocabulary size between the different datasets. Namely, in the DBpedia dataset each category is featured by a large number of words. This is expected, as the DBpedia articles are typically longer than the Freebase articles. The richest topics in DBpedia being *Religion*, *EntCult*, *TechIT*. In contrast, in the Freebase dataset the topics are being described by less words. The richest topics in Freebase are *Sports*, *TechIT*, *HumInt*. While for the Twitter dataset these topics are *Env*, *DisAcc*, *BusFi*.

Topic name	*Example tweets*
Business&Finance(*BusFi*)	visa cyber attack transactions affected account data safe company nbc
Disaster&Accident(*DisAcc*)	happening accident people dying could phone ambulance wakakkaka xd
Education(*Edu*)	read book even final pass pages period read
Environment(*Env*)	good complain cause part energized midterms happening
Entertainment&Culture(*EntCult*)	google adwords commercial greeeat enjoyed watching greeeeeat day
Health&Medical&Pharma(*Health*)	unprocessed fat eat lose fat real butter coconut oil eggs olive oil avocados raw nuts
Politics(*Pol*)	quoting military source sk media reports deployed rocket launchers decoys real
Sports(*Sports*)	ravens good position games left browns bengals playoffs
Technology&Internet(*TechIT*)	iphone cute ringtone download ringtone; lets enjoy wikileaks tomorrow publish direct message ever
War&Conflict(*War*)	nkorea prepared nuclear weapons holy war south official tells state media usa

Table 1. Example tweets for some of the evaluated topics after preprocessing (removing the stopwords, hastags, mentions and URLs).

When looking at the frequency of the entities in Figure 4, we can observe similar trends. The DBpedia articles contain the most number of entities for each topic, on average 22.24 ± 1.44 entities. From the full collection 69(0.72%) of the articles do not have any entity. In the case of Freebase, the average number of entities per article is

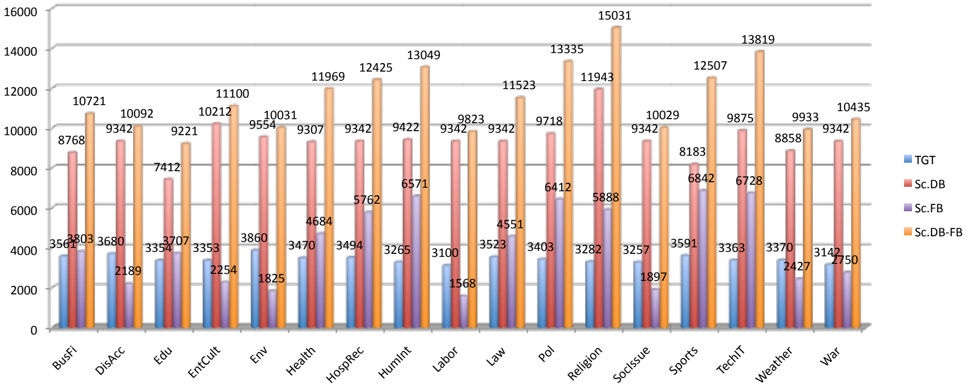


Fig. 3. The size of vocabulary in the source (Sc.DB, Sc.FB, Sc.DB-FB) and target (TGT) datasets after pre-processing.

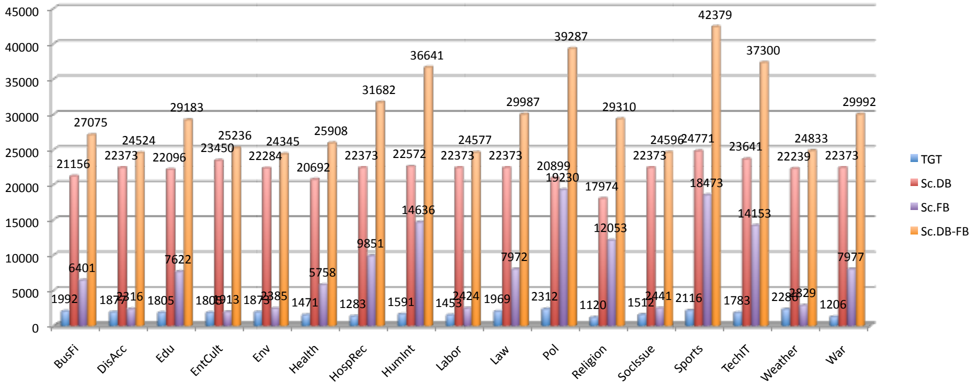


Fig. 4. The number of entities in the source (Sc. DB, Sc. FB, Sc. DB-FB) and target (TGT) datasets after pre-processing.

8.14 ± 5.78. The percentage of articles without any entity is 19.96%(3,377 examples). Lastly, the Twitter dataset contains the smallest number of entities, on average 1.73 ± 0.35 entities per articles. The number of articles mentioning no entity is 5,137 (41%).

The heatmap in Figure 5 demonstrates how the entity types' frequencies differ across different datasets. The darker the color, the higher the frequency of an entity in a dataset. According to this figure, Organization and Position have a relatively high frequency across all datasets. Other entities appearing frequently on these datasets are Person, Country and Natural Feature. Entity types such as MedicalCondition, or SportsEvent appear to be more representative of particular topics such as *Health* and *Sports*. When checking the clustering by topic in Figure 5, we can notice that the *Health* and *Edu* topics present a similar entity distribution in DBpedia and Freebase; the *War* topic

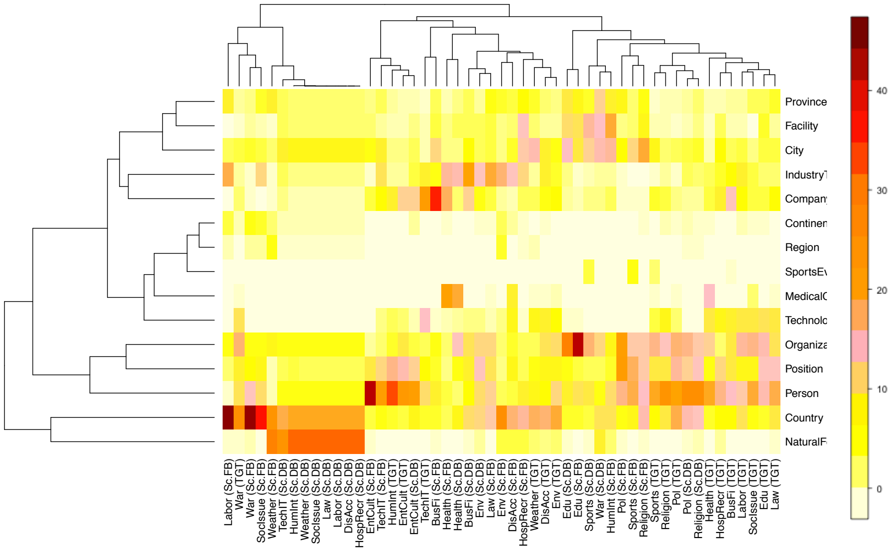


Fig. 5. The distribution of the top 15 entity types the in the source (Sc. DB, Sc. FB) and target (TGT) datasets.

has a similar entity distribution in Twitter and Freebase; while the *Pol* category presents a similar entity distribution in Twitter and DBpedia.

Based on the above figures on lexical richness and entity frequency, thus, we can notice that the Freebase dataset exhibits more similarity to Twitter than Dbpedia datasets. In order to get a better insight into this similarity, we will compare these datasets according to the proposed measures in the coming section.

6 Experiments

In this section we perform a series of experiments to investigate which KS exhibits more similarity to Twitter. In the first set of experiments we compare the performance of the SVM classifiers derived for the proposed cross-domain (CD) scenarios (Subsection 4.2), with the SVM classifier built on Twitter data only. These classifiers were trained using the different BoW and BoE features(Section 6.1) in each scenario. Therefore in this first set of experiments we address the questions of *which KS reflects better the lexical variation in Twitter?* and *what feature makes the KSs look more similar to Twitter?*.

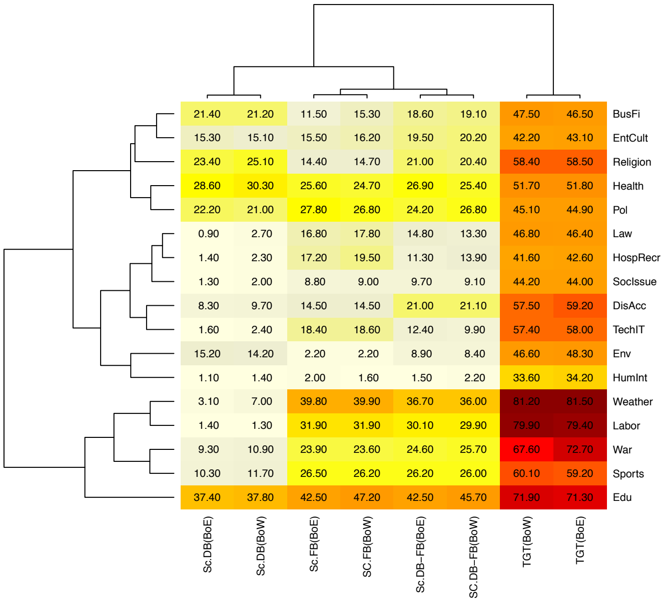
The second set of experiments, consists on computing the correlation between the proposed statistical measures (Section 6.2) and the accuracy of the CD classifiers. In this correlation analysis we investigate *which statistical measure presents the highest correlations with the accuracy of a CD classifier?* providing the most reliable estimate for the quality of KSs in topic classification of tweets.

6.1 Comparison of the Different Feature Sets in Cross-Domain Scenarios

The SVM classifiers derived from the CD scenarios –**Sc.DB**, **Sc.FB** and **Sc.DB-FB**– were evaluated based on their performance when trained using **BoW** and **BoE** features. The **TGT** SVM classifier –based on Twitter data only– was built on 80% of the Twitter data, and evaluated on 20% of the twitter data over five independent runs.

Figure 6 shows the results obtained using **BoW** and **BoE** features for the different CD scenarios. Based on the average performance in F1 measure, we can observe, that among the three CD scenarios, the best average performance was obtained by the **Sc.DB-FB** SVM classifier using **BoW** features, which is followed by the **Sc.FB** and **Sc.DB** SVM classifiers also using **BoW** features.

Using both feature sets, we found that for the **Sc.DB-FB** scenario the topics which presented a performance closer to the one obtained by the **TGT** classifier were the *Weather* and *Edu*. For the **Sc.FB** scenario these topics were the *Edu*, *Weather*, *Labor*. Finally for the **Sc.DB** scenario these topics were the *Edu*, *Health*. The topics for which the performance was higher using **BoE** features were the *BusFi*, *Env*, *Pol*, *SocIssue*, *Sports*. For *Labor* the performance was the same for both features .



Sc.DB(BoE)	Sc.DB(BoW)	Sc.FB(BoE)	SC.FB(BoW)	Sc.DB–FB(BoE)	SC.DB–FB(BoW)	TGT(BoW)	TGT(BoE)	
21.40	21.20	11.50	15.30	18.60	19.10	47.50	46.50	BusFi
15.30	15.10	15.50	16.20	19.50	20.20	42.20	43.10	EntCult
23.40	25.10	14.40	14.70	21.00	20.40	58.40	58.50	Religion
28.60	30.30	25.60	24.70	26.90	25.40	51.70	51.80	Health
22.20	21.00	27.80	26.80	24.20	26.80	45.10	44.90	Pol
0.90	2.70	16.80	17.80	14.80	13.30	46.80	46.40	Law
1.40	2.30	17.20	19.50	11.30	13.90	41.60	42.60	HospRecr
1.30	2.00	8.80	9.00	9.70	9.10	44.20	44.00	SocIssue
8.30	9.70	14.50	14.50	21.00	21.10	57.50	59.20	DisAcc
1.60	2.40	18.40	18.60	12.40	9.90	57.40	58.00	TechIT
15.20	14.20	2.20	2.20	8.90	8.40	46.60	48.30	Env
1.10	1.40	2.00	1.60	1.50	2.20	33.60	34.20	HumInt
3.10	7.00	39.80	39.90	36.70	36.00	81.20	81.50	Weather
1.40	1.30	31.90	31.90	30.10	29.90	79.90	79.40	Labor
9.30	10.90	23.90	23.60	24.60	25.70	67.60	72.70	War
10.30	11.70	26.50	26.20	26.20	26.00	60.10	59.20	Sports
37.40	37.80	42.50	47.20	42.50	45.70	71.90	71.30	Edu

Fig. 6. The performance in F1 measure of the Sc.DB, Sc.FB, Sc.DB-FB and TGT classifiers using **BoW** and **BoE** features for each topic over five independent runs. The training set of TGT classifier consists of 80% of the Twitter dataset (9,928 tweets). The Sc.DB, Sc.FB and Sc.DB-FB classifier were trained only on social knowledge sources data.

A slightly different trend can be observe for the **TGT** classifier, where the best average F1 measure was achieved using **BoE** features. There were 10 topics for which

BoE features were useful: *DisAcc*, *EntCult* , *Env*, *Health*, *HospRec*, *HumInt*, *Religion*, *TechIT*, *War* and *Weather*.

Overall, our results indicate that **Sc.FB** KS is more similar to Twitter than **Sc.DB**. Furthermore, combining the contribution of the **Sc.DB** and **Sc.FB** is beneficial for detecting topics in Tweets, since the **Sc.DB-FB** scenario achieves the best overall results. With regard to the features, we found that in 11 out of 17 cases the results obtained using **BoW** features were better, and in 5 out of 17 cases the **BoE** features were found more effective.

We also compared the performance of the Twitter classifier against the three CD classifiers over the full learning curve, by gradually increasing the number of tweets used to train the classifier. Our analysis revealed that in the majority of the cases the CD classifiers worked relatively well. That is, a sufficient amount of annotated tweets were needed to significantly outperform the three CD classifiers over the full learning curve. The number of annotations needed for each topic is summarised in Table 2. For e.g. for more than 9 out of 17 topics the necessary amount of annotated tweets need to exceed 900. However, in a real-world scenario annotating tweets is an expensive task.

BusFi	*DisAcc*	*Edu*	*EntCult*	*Env*	*Health*	*HospRec*	*HumInt*	*Labor*
$993^{\clubsuit}$	$993^{\clubsuit}$	$993^{\clubsuit}$	$993^{\clubsuit}$	$993^{\clubsuit}$	$1,986^{\clubsuit}$	$1,986^{\spadesuit}$	$160^{\spadesuit}$	$320^{\spadesuit}$
Law	*Pol*	*Religion*	*SocIssue*	*Sports*	*TechIT*	*Weather*	*War*	
$640^{\spadesuit}$	$993^{\clubsuit}$	$320^{\spadesuit}$	$320^{\spadesuit}$	$993^{\spadesuit}$	$640^{\clubsuit}$	$320\spadesuit$	$640^{\clubsuit}$	

Table 2. Number of annotated tweets required for the Twitter classifier to beat the Sc.DB, Sc.FB and Sc.DB-FB CD classifiers. Significance levels: p-value $< \clubsuit 0.01 \spadesuit 0.05$

6.2 Comparison of Statistical Measures in Topic Classification of Tweets

In this second set of experiments we aimed to investigate our research question of *how similar or dissimilar are social knowledge sources to Twitter posts; and which similarity measure does better reflect the lexical changes between KSs and Twitter posts?*. We thus performed a comparison between the proposed KL divergence, $cosine$ similarity and χ^2 test by measuring the correlation of these values with the performance of a CD classifier using Sc.DB, Sc.FB and Sc.DB-FB scenarios.

Each CD classifier was evaluated on 20% of the Twitter data, and the performance was averaged over five independent runs. The obtained F1 measures for the CD classifiers were then compared with the values obtained for the different statistical measures, and the Pearson correlation was computed.

Figure 7 show the correlations obtained using KL, cosine and χ^2 values. A positive correlation indicates that the performance of the CD classifiers increases as the divergence decreases (the distribution are more similar); while a negative correlation indicates that the performance increases as the divergence increases (the distributions are less similar). As we can notice, for the KL scores, there are 24 cases in which the correlation scores are higher than 70% in absolute terms. In the case of $Cosine$ similarity these cases sum up to 25. While in the case of χ^2 values for a total of 32 cases were the correlation values higher than 70%.

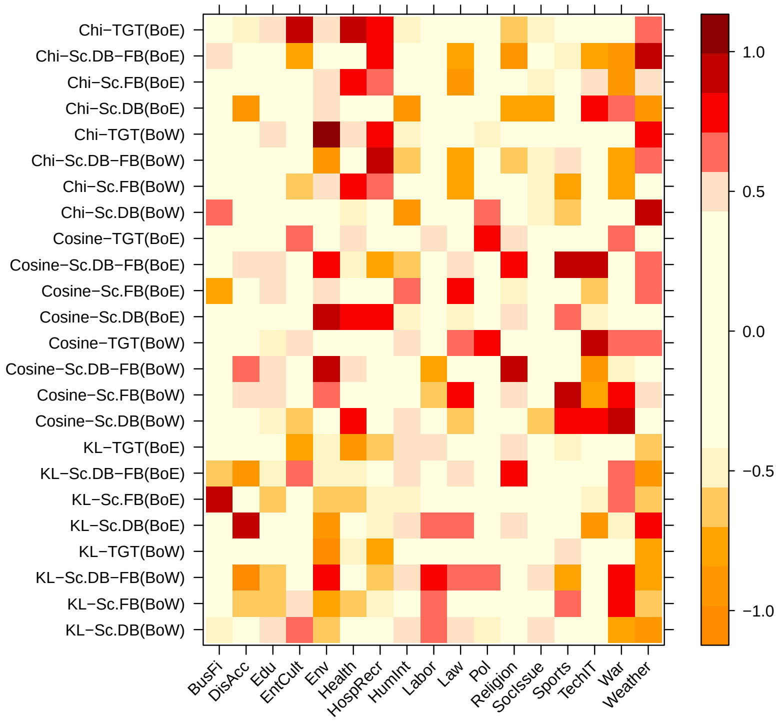


Fig. 7. The Pearson correlation between the performance in F1 of the Sc.DB, Sc.FB, Sc.DB-FB CD classifiers and the KL, Cosine and χ^2 measures

Based on these results, we found the χ^2 to provide the best correlation scores for the usefulness of the KSs data. The second best score was for the $cosine$ similarity, which was followed by the KL measure.

Figure 8 shows the pairwise similarity obtained for the source and target datasets according to $(\chi^2)^{-1}$ similarity measure.[11] As expected the closest datasets to the test Twitter dataset is the training set for the Twitter classifier (ChiSc.TGT). The second closest dataset according to χ^2 is the Sc.Fb dataset. The Sc.DB and Sc.DB-FB are then the less similar datasets to the test dataset.

7 Conclusions and Future Work

In this paper we presented a first attempt towards understanding the usefulness of DBpedia and Freebase KSs in CD topic classification of tweets. We presented an analysis

[11] As χ^2 measure distance rather than similarity we inverted its value to present the similarity between topics better.

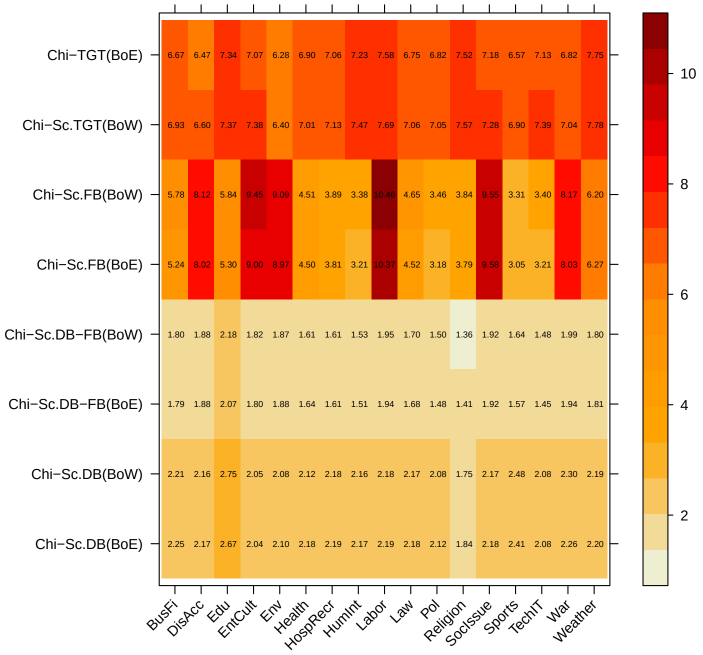


Fig. 8. The values of $(\chi^2)^{-1} * 10^{-5}$ for each Sc.DB, Sc.FB, Sc.DB-FB, TGT scenarios. High values indicate that the topics are more similar between the source and target dataset.

between these data sources focusing on various lexical features (**BoW**) and entity features(**BoE**).

For a total of 17 topics we compiled a gold standard for each individual KS, and for the joint set of these sources. From the resulted datasets we then built three CD classifiers which we evaluated against a Twitter classifier using the different features.

Our analysis revealed that from the two KSs, Freebase topics seem to be much closer to the Twitter topics than the DBpedia topics due to the much restricted vocabulary used in Freebase. Furthermore, we found that the two KSs contain complementary information, i.e.; the joint dataset was found more useful than the individual KS datasets. With regard to the feature sets, we found that for the three CD classifiers on average the results obtained using **BoW** were better than those obtained with **BoE** in 5 out of 17 cases.

When comparing the results of these CD classifiers to the Twitter classifier we found that for some of the topics the Twitter classifier required a large number of annotations to outperform these classifiers, indicating that in the absent of any annotated tweets, applying these CD classifiers is still beneficial. Previous research on transfer learning has also shown, that outperforming the target (Twitter) classifier is extremely difficult for many tasks including sentiment classification ([5, 13]). A promising alternative found in the literature was to combine the annotated examples in the source and target datasets([6]). Our future work aims to follow this direction, focusing on building transfer learning algorithms which can effectively combine the contribution of the two KSs; and also exploring other features derived from the named entities.

Finally, we also looked at various statistical measures for predicting the usefulness of the data gathered from these KSs. These experiments revealed the χ^2 test as being the best measure for quantifying the distributional differences among between KSs and Twitter. Our future work in this direction will focus in investigating more accurate measures for quantifying this difference for e.g. by taking into account the special vocabulary (abbreviations, misspellings, shortening) used in Twitter, and normalise this to standard English terms ([8]).

References

1. Identifying topics in social media posts using dbpedia. In O. e. a. Munoz-Garcia, editor, *In Proceedings of the NEM Summit (27-29 September 2011)*, pages 81–86, 2011.
2. F. Abel, Q. Gao, G.-J. Houben, and K. Tao. Analyzing user modeling on twitter for personalized news recommendations. In *Proceedings of the 19th international conference on User modeling, adaption, and personalization*, UMAP'11, pages 1–12, Berlin, Heidelberg, 2011. Springer-Verlag.
3. B. Bigi. Using kullback-leibler distance for text categorization. In *Advances in Information Retrieval, Lecture Notes in Computer Science Volume 2633*, 2003.
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. *J. Web Sem.*, 7(3):154–165, 2009.
5. J. Blitzer, M. Dredze, and F. Pereira. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Association for Computational Linguistics*, Prague, Czech Republic, 2007.
6. H. Daume III. Frustratingly easy domain adaptation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*.
7. P. Ferragina and U. Scaiella. Tagme: on-the-y annotation of short text fragments (by wikipedia entities). In *Proc of the CIKM'10*, 2010.
8. B. Han and T. Baldwin. Lexical normalisation of short text messages: makn sens a #twitter. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 368–378, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.
9. S. P. Kasiviswanathan, P. Melville, A. Banerjee, and V. Sindhwani. Emerging topic detection using dictionary learning. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, CIKM '11, pages 745–754, New York, NY, USA, 2011. ACM.
10. E. Meij, W. Weerkamp, and M. de Rijke. Adding semantics to microblog posts. In *Proceedings of the fifth ACM international conference on Web search and data mining, WSDM '12*.
11. D. Milne and I. H. Witten., editors. *Learning to link with Wikipedia.* 2008.
12. P. K. P. N. Mendes, A. Passant and A. P. Sheth. Linked open social signals. In *In WI-IAT 10*, 2010.
13. S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, October 2010.