Semantic Network-driven News Recommender Systems: a Celebrity Gossip Use Case

Marco Fossati, Claudio Giuliano, and Giovanni Tummarello
{fossati,giuliano,tummarello}@fbk.eu

Fondazione Bruno Kessler, via Sommarive 18, 38123 Trento, Italy

Abstract. Information overload on the Internet motivates the need for filtering tools. Recommender systems play a significant role in such a scenario, as they provide automatically generated suggestions. In this paper, we propose a novel recommendation approach, based on semantic networks exploration. Given a set of celebrity gossip news articles, our systems leverage both natural language processing text annotation techniques and knowledge bases. Hence, real-world entities detection and cross-document entity relations discovery are enabled. The recommendations are enhanced by detailed explanations to attract end users' attention. An online evaluation with paid workers from crowdsourcing services proves the effectiveness of our approach.

Keywords: Data Integration, Natural Language Processing, Information Retrieval, Information Filtering, Entity Linking, Recommendation Strategy

1 Introduction

The amount of publicly available data on the World Wide Web nowadays has dramatically increased and has led to the problem of information overload. Recommender systems try to tackle this issue by offering personalized suggestions. News recommendation is a real-world application of such systems and is growing as fast as the online news reading practice: it is estimated that, in May 2010, 57% of U.S. Internet users consumed online news by visiting news portals [7]. Recently, online news consumers seem to have changed the way they access news portals: "just a few years ago, most people arrived at our site by typing in the website address. (...) Today the picture is very different. Fewer than 50% of the 8 million+ visitors to the News website every day see our front page and the rest arrive directly at a story", a product manager of the BBC News website affirms,[1] indicating the need for news information filtering tools.

The online reading practice leads to the so-called *post-click* news recommendation problem: when a user has clicked on a news link and is reading an article, he or she is likely to be interested in other related articles. This is still a typical editor's task, namely an expert who manually looks for relevant content and

[1] http://www.bbc.co.uk/blogs/bbcinternet/2012/03/bbc_news_facebook_app.html

builds a recommendation set of links, which will be displayed below or next to the current article. The primary aim is to keep users navigating on the visited portal. News recommender systems attempt to automate such task. Current strategies can be clustered into 3 main categories [5], namely (a) collaborative filtering, (b) content-based recommendation, and (c) knowledge-based recommendation. (a) focuses on the similarities between users of a service, thus relying on user profiles data. (b) leverages term-driven information retrieval techniques to compute similarities between items. (c) mines external data to enrich item descriptions.

In this paper, we propose a novel news recommendation strategy, which leverages both natural language processing techniques and semantically structured data. We show that entity linking tools can be coupled to existing knowledge bases in order to compute unexpected suggestions. Such knowledge bases are used to discover meaningful relations between entities. As a preliminary work to assess the validity of our approach, we focus on a celebrity gossip use case and consume data from the TMZ news portal and the Freebase graph database.[2] For instance, given a TMZ article on `Michael Jackson`, our strategy is able to detect from Freebase that `Michael Jackson` (a) is a dead celebrity who had drug problems and (b) dated with `Brooke Shields`, thus suggesting other TMZ articles on `Amy Winehouse`, `Kurt Cobain` (other dead celebrities who had drug problems) and `Brooke Shields`. We investigate if user attention can be attracted via specific explanations, which clarify why a given recommendation set is proposed. Such explanations are built on top of the entity relations. Finally, we conducted an online evaluation with real users. We outsourced a set of experiments to the community of paid workers from Amazon's Mechanical Turk (AMT) crowdsourcing service.[3] The collected results confirm the effectiveness of our approach.

Our primary aim is to attract the attention of a generic user, since post-click news recommendation generally relies on a single click user profile data. Therefore, we are set apart from most traditional recommender systems with respect to three main features:

1. *User agnosticity*: user interests are deduced from user profile data and contribute to the quality of recommendations. Collecting explicit feedback is a costly task, as it requires motivated users. Our approach gives low priority to user profiles.
2. *Unexpectedness*: similarity, novelty and coherence are key components for satisfactory news recommendations [7]. Content-based strategies tend to propose too similar items and create an 'already seen' sensation. We believe entity relations discovery can augment both novelty and coherence, thus leading to unexpected suggestions.
3. *Specific explanation*: in news web portals, generic sentences such as `Related stories` or `See also` are typically shown together with the recommendation set. We expect that more specific sentences can improve the trustworthiness of the system.

[2] `http://www.tmz.com`, `http://www.freebase.com/`
[3] `https://www.mturk.com/mturk/welcome`

2 Related Work

Content-based recommendation applies to unstructured text, such as news articles. Document representation with bag-of-words vector space models and the cosine similarity function still represent a valid starting point to suggest topic-related documents [11]. Knowledge extraction from structured data is an attested knowledge-based strategy. Linked Open Data (LOD) datasets, e.g., DBpedia[4] and Freebase are queried to enrich with properties the entities extracted from news articles [6], to collect movie information for movie schedules recommendations [12], or to suggest music for photo albums [1]. Structured data may be also mined in order to compute similarities between items, then between user and items [5]. Content-based and knowledge-based approaches must be combined into hybrid systems in order to achieve better results. Lašek [6] proposes a hybrid news articles recommendation system, which merges content processing techniques and data enrichment via LOD.

Recommender systems evaluation frameworks boil down to two main approaches [5], namely (a) offline and (b) online. (a) leverages gold-standard datasets and aims at estimating the performance of a recommendation algorithm via statistical measures. (b) relies on real user studies. Ziegler et al. [13] adopt both approaches. Hayes et al. [4] argue that user satisfaction corresponds to the actual use of a system and can be effectively measured only via online evaluation. The interest in exploiting crowdsourcing services for dataset building and online evaluation has recently grown, especially with respect to natural language processing tasks [10] and behavioral research [8].

3 Approach

Our strategy merges content-based and knowledge-based approaches and is defined as a *hybrid entity-oriented* recommendation strategy enhanced by human-readable explanations. Given a source article from a news portal, we recommend other articles from the portal archive, namely the corpus, by leveraging both entity linking techniques and knowledge extraction from semantically structured knowledge bases. Specifically, we gathered a celebrity gossip corpus from TMZ and chose Freebase as the knowledge base.

We consider both the corpus and the knowledge base as a unique object, namely a *dataspace*, which results from heterogeneous data sources integration. Each data source is converted into an RDF graph and becomes an element of the dataspace. Such dataspace can then be queried in order to retrieve sets of recommendations. A *semantic recommender* exploits SPARQL graph navigation capabilities to output recommendation sets. Each recommender is built on top of a concept, e.g., *substance abuse*.

The entity linking step in the corpus processing phase enables the detection of both real-world entities and encyclopedic concepts. We compute concept statistics on the whole corpus and assume that the most frequent ones are likely to generate interesting recommendations. A mapping between corpus concepts and meaningful relations of the knowledge base allows the creation of recom-

[4] `http://dbpedia.org/`

menders. Table 1 shows the TMZ-to-Freebase n-ary concept mapping we manually built. Each Freebase value represents the starting point for the construction of a recommender, while the string after the last dot becomes the name of the recommender, e.g., *parents*.

Given an entity of the source article, a name of a recommender and an entity contained in the recommendation sets, we are able to construct a specific explanation. Ultimately, a ranking of all the recommendation sets produces the final top-N suggestions output.

Table 1: TMZ-to-Freebase mapping

TMZ	Freebase
Family	people.person.{parents, sibling_s, children, spouse_s}
Intimate_relationship	celebrities.celebrity.sexual_relationships
Dating	base.popstra.celebrity.dated
Ex_(relationship)	base.popstra.celebrity.breakup
Net_worth	celebrities.celebrity.net_worth
Substance_abuse	celebrities.celebrity.substance_abuse_problems
Conviction	base.crime.convicted_criminal
Court	law.court.legal_cases
Arrest	base.popstra.celebrity.{arrest, prison_time}
Legal_case	law.legal_case.subject
Criminal_charge	celebrities.celebrity.legal_entanglements
Judge	law.judge
Death	people.deceased_person
Television_program	tv.tv_program

4 System Architecture

Figure 1 describes the general system workflow. The major phases are (a) corpus processing, (b) knowledge base processing, (c) dataspace querying and (d) recommendation ranking.

TMZ Processing Pipeline. Given as input a set of TMZ articles, we output an RDF graph and load it into the dataspace. Corpus documents are harvested via a subscription to the TMZ RSS feed. The RSS feed returns semi-structured XML documents. A cleansing script extracts raw text from each XML document. The entity linking step exploits *The Wiki Machine*,[5] a state-of-the-art [9] machine learning system designed for linking text to Wikipedia, based on a word sense disambiguation algorithm [2]. For each raw text document, real-world entities such as persons, locations and organizations are recognized, as well as encyclopedic concepts. This enables (a) the assignment of a unique identifier, namely a DBpedia URI to each annotation and (b) the choice of top corpus concepts for recommenders building purposes. The Wiki Machine takes a plain text as input and produces an RDFa document.[6] The extracted terms are assigned an `rdf:type`, namely `NAM` for real-world entities or `NOM` for encyclopedic concepts. The `hasLink` property connects the terms to the article URL they belong, thus enabling the computation of the recommendation set. Other metadata, such as

[5] `http://thewikimachine.fbk.eu`
[6] The full corpus of TMZ RDFa documents is available at `http://bit.ly/QLph9B`

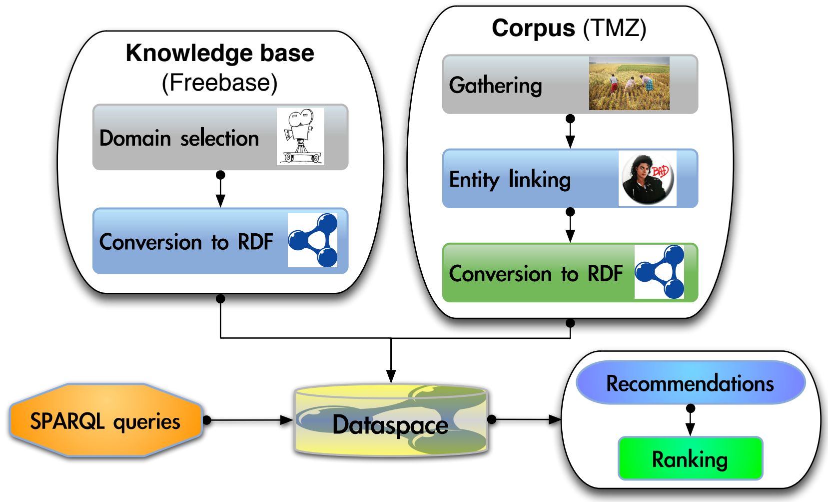

Fig. 1: High level system workflow

the link to the corresponding Wikipedia page and the annotation confidence score are also expressed. RDFa documents are converted into RDF data via the Any23 library.[7] RDF data is loaded into a Virtuoso[8] triple store instance, which serves the dataspace for querying.

Freebase Processing Pipeline. Freebase provides exhaustive granularity for several domains, especially for celebrities. Given that such knowledge base is large, we avoid loading its complete version, because of severe performance issues we encountered. Consequently, meaningful slices corresponding to the corpus domains, e.g., celebrities, people, are selected. A domain-dependent subset is then produced via a filter written in Java. The dataset is converted into RDF data with logic implemented in Java. Finally, RDF data is loaded into a Virtuoso triple store instance.

4.1 Querying the Dataspace

A recommender performs a join between an entity belonging to the TMZ graph and the corresponding entity belonging to the Freebase graph. TMZ entities are identified by a DBpedia URI, which differs from the Freebase one. Therefore, we exploit `sameAs` links between DBpedia and Freebase URIs. Recommenders are divided in two categories, namely (a) *entity-driven* and (b) *property-driven*.[9] For each detected entity of the source article, we run Freebase schema inspection queries[10] and retrieve its types and properties. Thus, we are able to recognize which recommenders can be triggered for a given entity. Building a recommender

[7] `http://incubator.apache.org/any23/`
[8] `http://virtuoso.openlinksw.com/`
[9] The full sets are available at `http://bit.ly/MWGu06` and `http://bit.ly/MWGsW3`
[10] Available at `http://bit.ly/MVGVtE`

requires (a) knowledge of relevant Freebase schema parts in order to properly browse its graph and (b) a sufficiently expressive RDFa model for named entities and link retrieval. The `NAM` type and the `hasLink` property provide such expressivity.

Entity-Driven Recommenders. The queries behind entity-driven recommenders contain an `%entity%` parameter that must be programmatically filled by an entity belonging to the source article. For instance, given an article in which `Jessica Simpson` is detected and triggers the *sexual relationships* recommender, we are able to return all the corpus articles (if any) that mention entities who had sexual relationships with her, e.g., `John Mayer`. To avoid running empty-result recommenders, we built a set of `ASK` queries,[11] which check if recommendation data exists for a given entity. The *sexual relationships* query follows:

```
PREFIX fb: <http://rdf.freebase.com/ns/>
PREFIX twm: <http://thewikimachine.fbk.eu#>
SELECT DISTINCT ?had_relationship_with ?link
WHERE  <http://dbpedia.org/resource/%entity%> owl:sameAs ?fb_entity .
?fb_entity fb:celebrities.celebrity.sexual_relationships ?fb_sexual_rel .
?fb_sexual_rel fb:celebrities.romantic_relationship.celebrity ?fb_celeb .
?fb_celeb fb:type.object.name ?had_relationship_with .
?dbp_celeb owl:sameAs ?fb_celeb ; a twm:NAM ; twm:hasLink ?link ; twm:hasConfidence ?conf .
FILTER (?fb_entity != ?fb_celeb) . FILTER (lang(?had_relationship_with)='en') .
ORDER BY DESC (?conf)
```

Property-Driven Recommenders. After the schema inspection step, an entity of the source article can directly trigger one of these recommenders if it contains the corresponding property. Property-driven queries return articles that mention entities who share the same property. Hence, they do not require a parameter to be filled. For instance, given an article in which `Lindsay Lohan` is detected and the property `legal entanglements` is identified during the schema inspection step, we can suggest other articles on people who had legal entanglements, e.g., `Britney Spears`.

Building Explanations. Specific explanations are handcrafted from $<s, r, o>$ triples, where s is a *subject* entity that was extracted from the source article, r is the *relation* expressed by the triggered recommender and o is an *object* entity for which the recommendation set is computed. Therefore, we are able to construct different explanations depending on the elements we use. For instance, (a) s,r,o yields: `Jessica Simpson` had `sexual relationships` with `John Mayer`. Read more about him. (b) s,r yields: Read more about `Jessica Simpson`'s `sexual relationships`. (c) r,o yields: Read more about her `sexual relationships` with `John Mayer`.

4.2 Ranking the Recommendation Sets

Since recommendations originate from database queries, they are unranked and in some cases too many. To overcome the problem, we implemented an information retrieval ranking algorithm and are able to provide top-N recommendations.

[11] Available at `http://bit.ly/NDNORH`

The bag-of-words (BOW) cosine similarity function is known to perform effectively for topic-related suggestions [11]. However, it does not take into account language variability. Consequently, we also leverage a latent semantic analysis (LSA) algorithm.[12] The final score of each corpus article is the sum of BOW and LSA scores and is assigned to the article URL. Afterwards, we run all the recommenders and intersect their result sets with the BOW+LSA ranking of the whole corpus, thus producing a so-called *semantic* ranking. This represents our final output, which consists of a ranked set of article URLs associated to the corresponding recommenders names.

5 Evaluation

The assessment of end user satisfaction has high priority in our work. According to Hayes et al. [4], we consequently decided to adopt an online evaluation approach with real users. In this scenario, the major issue consists of gathering a sufficiently large group of people who are willing to evaluate our systems. Crowdsourcing services provide a solution to the problem, as they allow us to outsource the evaluation task to an already available massive community of paid workers. To the best of our knowledge, no news recommender systems have been evaluated with crowdsourcing services so far. We set up an experimental evaluation framework for AMT, via the CrowdFlower platform.[13] A description of the mechanisms that regulate AMT is beyond the scope of the present paper: the reader may refer to [8] for a detailed analysis.

Our primary aim is to demonstrate that evaluators generally prefer our recommendations. Thus, we need to put our strategy in competition with a baseline. We leveraged the already implemented BOW+LSA information retrieval ranking algorithm. In addition, we set two specific objectives, related to the *specific explanation* and *unexpectedness* assumptions, as outlined in Section 1: (a) confirm that a specific explanation better attracts user attention rather than a generic one; (b) check if the recommended items are interesting, although they may appear unrelated and no matter what kind of explanation is provided.

Quality control of the collected judgements is a key factor for the success of the experiments. The essential drawback of crowdsourcing services relies on the cheating risk: workers (from now on called *turkers*) are generally paid a few cents for tasks which may only need a single click to be completed. Hence, it is highly probable to collect data coming from random choices that can heavily pollute the results. The issue is resolved by adding *gold* units, namely data for which the requester already knows the answer. If a turker misses too many gold answers within a given threshold, he or she will be flagged as untrusted and his or her judgments will be automatically discarded.

5.1 General Setting

Our evaluation framework is designed as follows: (a) the turker is invited to read a complete news article. (b) A set of recommender systems are displayed

[12] `http://hlt.fbk.eu/en/technology/jlsi`
[13] `http://crowdflower.com/`

below the article. Each system consists of a natural language *explanation* and a news title *recommendation*. (c) The turker is asked to give a preference on the most attracting recommendation, namely the one he or she would click on in order to read the suggested article. A single experiment (or *job*) is composed of multiple data *units*. A unit contains the text of the article and the set of explanation-recommendation pairs. Figure 2 shows a unit fragment of the actual web page that is given to a turker who accepted one of our evaluation jobs. Both instructions and question texts need to be carefully modeled, as they must mirror the main objective of the task and should not bias turkers' reaction. Since we aim at evaluating user attention attraction, we formulated them as per Figure 2.

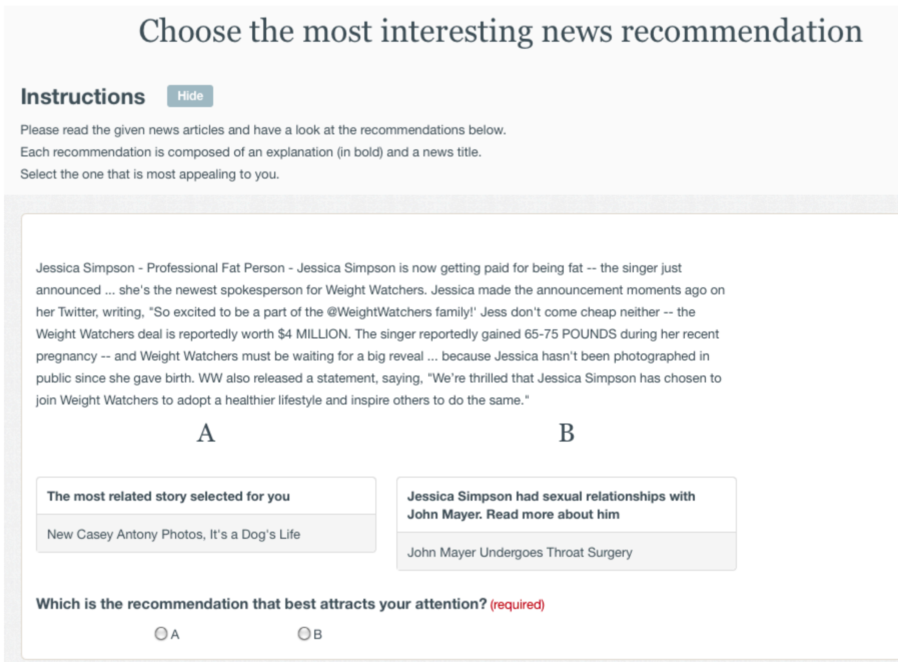

Fig. 2: Web interface of an evaluation job unit

5.2 Experiments

Table 2 provides an overview of our experimental environment. The parameters we have isolated for a single experiment are presented in Table 2a. On top of the possible variations, we built a set of nine experiments, which are described in Table 2b. We modeled two Q values, namely direct (as per Figure 2) and indirect (`Which recommendation do you consider to be more trustworthy?`), to monitor a possible alteration of turkers' reaction. Experiments having $A = 5$ aim at decreasing the probability a turker gets trusted by chance, because he or she accidentally selected correct gold answers. They have an additional F value in the *Rec* parameter, as we randomly extracted 3 fake recommendations per unit from a file with more than 2 million news titles. However, such an architectural choice generated noisy results, since it occurred that some fake titles were selected.[14] *Exp* is a key parameter, which allows us to check whether the presence or the absence of a specific explanation represents a discriminating factor. *SExp* is intended to measure the effectiveness of a specific explanation while reducing its complexity.

Table 2: Experiments overview

(a) Parameters

Parameter	**Values**
Q	D, I
A	B, M
Exp	GS, G
SExp	SRO, SR, RO, R
Rec	B, S, F

(b) Configuration

Name	**Q**	**A**	**Exp**	**SExp**	**Rec**
Pilot	D	2	GS	SRO	B, S
Same explanation	D	2	G	None	B, S
4 generic + 1 specific	D	5	GS	SRO	B, S, F
5 generic	D	5	G	None	B, S, F
Same recommendation	D	2	GS	SRO	S
Relation only	D	5	GS	R	B, S, F
Subject + relation	D	5	GS	SR	B, S, F
Object + relation	D	5	GS	RO	B, S, F
Indirect	I	2	GS	SRO	B, S

Legend

Q	Question
A	Answer
Exp	Explanation
SExp	specific explanation
Rec	Recommendation

D	Direct
I	Indirect
2	Binary
5	5 choices
GS	Generic + specific
G	Generic only

SRO	Subject + relation + object
SR	Subject + relation
RO	Relation + object
R	Relation only
B	Baseline
S	Semantic
F	Fake

Each job contains 8 regular + 2 gold units, namely 5 articles proposed twice, in combination with 2 significant (and eventually 3 fake) explanation-recommendation pairs. The recommendation titles of the regular units are extracted from the top-2 links of the baseline and the semantic rankings. Gold is created by extracting the title from the last, i.e., less related link of the baseline ranking, the top link of the semantic ranking and assigning the correct answer to the latter. We collected a minimum of 10 valid judgments per unit and set the number of units per page to 3.

Once the results obtained, it frequently occurred that the expected number of judgments was higher: depending on their accuracy in providing answers to gold units, turkers switched from untrusted to trusted, thus adding free extra judgments. The proposed articles come from the TMZ website, which is well known in the United States. Therefore, we decided to gather evaluation data only from American turkers. The total cost of each experiment was 3.66$.

After visiting some news web portals, we chose the following generic explanations and randomly assigned them to both the baseline and the fake recommendations: (a) `The most related story selected for you`; (b) `If you liked`

[14] See Table 3 for further details.

Table 3: Absolute results per experiment. ♢, ♠ and ♣ respectively indicate statistical significance differences between baseline and semantic methods, with $p < 0.05$, $p < 0.01$ and $p < 0.001$

Experiment	**Judgments**	**Fake %**	**Baseline %**	**Semantic %**
Pilot 1	82	0	40.24	**59.76**♢
Pilot 2	80	0	32.5	**67.5**♠
Same explanation	80	0	48.75	51.25
4 generic + 1 specific	90	3.33	23.33	**73.33**♣
5 generic	88	13.63	37.5	48.86
Same recommendation	86	0	36.04	**63.96**♠
Relation only	68	13.23	41.17	45.58
Indirect	82	0	37.8	**62.2**♠
Subject + relation	86	8.13	41.86	50
Object + relation	68	5.88	41.17	52.94

`this article, you may also like`; (c) `Here for you the hottest story from a similar topic`; (d) `More on this story`; (e) `People who read this article, also read`. 2 regular units were removed from the *relation only* and the *object + relation* experiments: it was impossible to build specific explanations with an implicit subject or object, since the entities that triggered the recommendations differed from the main entity of the source article.

5.3 Results

Table 3 provides an aggregated view of the results obtained from the Crowdflower platform.[15] With respect to the absolute percentage values, we first observe that our approach always outperformed the baseline. Furthermore, statistical significance differences emerge when a complete $<s, r, o>$ specific explanation is given. We ran twice, i.e., in two separate days the *pilot* experiment and noticed an improvement. The *indirect* experiment only differs from the pilot in the question parameter and yielded similar results. The *4 generic + 1 specific* experiment has the highest semantic percentage: this translates into an expected behavior, since the presence of a single specific explanation against four generic ones is likely to bias turkers' reaction towards our approach. As the complexity of the specific explanation decreases, i.e., in the *subject + relation, object + relation* and *relation only* experiments or when only generic explanations are presented, namely in the *5 generic* and *same explanation* experiments, judgments towards our approach tend to decrease too. Hence, we evince the importance of providing specific explanations in order to attract user attention.

5.4 Discussion

Experiments containing a specific explanation aim at assessing its attractive power (assumption 3). If we compare experiments which only differ in the *Exp* parameter, namely *4 generic + 1 specific* and *5 generic*, *pilot 1-2* and *Same explanation*, in the formers turkers prefer our strategy with a statistically sig-

[15] The complete set of full reports is available at `http://bit.ly/MOrN30`

nificant difference. Therefore, specific explanations are proven to enhance the trustworthiness of the system.

The evaluation of the unexpectedness factor (assumption 2) boils down to check whether turkers privilege the novelty of a recommendation or its similarity to the source article. In experiments including only generic explanations, namely *Same explanation* and *5 generic*, we noticed the following: (a) no statistically significant differences exist between the strategies; (b) when the baseline returns articles that are unrelated to the topic or the entity of the source article, turkers prefer our strategy and vice versa. Hence, we argue that users tend to privilege similarity if they are given a generic explanation. On the other hand, when the baseline strategy suggests a clearly related article and when a specific explanation is provided, turkers tend to choose our strategy even if it suggests an apparently unrelated article. This is a first proof of the unexpectedness factor: users are attracted by the specific explanation and are eager to read an unexpected article rather than another article on the same topic/entity.

6 Conclusion

In this paper, we presented a novel recommendation strategy leveraging entity linking techniques in unstructured text and knowledge extraction from structured knowledge bases. On top of it, we build hybrid entity-oriented recommender systems for news filtering and post-click news recommendation. We argued that entity relations discovery leads to unexpected suggestions and specific explanations, thus attracting user attention. The adopted online evaluation approach via crowdsourcing services assessed the validity of our systems. A demo prototype consumes Freebase data to recommend TMZ celebrity gossip articles and can be viewed at `http://spaziodati.eu/widget_recommendation/`. For our future work, we have set the following milestones:

1. *Ecological evaluation.* AMT allowed us to build fast and cheap online evaluation experiments. However, the collected judgments may be biased by the politeness effect of the economical reward and the turkers' awareness of performing a question-answering task. Therefore, we intend to set up an ecological evaluation scenario, which simulates a real-world usage of our recommender systems and enables natural user reactions. We will adopt the Google AdWords[16] approach proposed by Guerini et al. [3].
2. *Methodology for building recommenders.* Currently, we have manually implemented a domain-specific list of recommenders, based on the most frequent corpus concepts. We plan to automate this process by extracting generic relations from Freebase via data analytics techniques.
3. *Methodology for building specific explanations.* Explanations are naively mapped to the relations and the corresponding subject/object entities. How to automatically build linguistically correct sentences remains an open problem.
4. *User profile construction.* Explicit and implicit user preferences acquisition can improve the quality of the recommendations. Our demo page may serve

[16] `http://adwords.google.com/`

as a platform for gathering such data. Otherwise, we may adapt our systems to datasets containing user ratings.

Acknowledgements. This work was supported by the EU project Eurosentiment, contract number 296277.

References

1. Chao, J., Wang, H., Zhou, W., Zhang, W., Yu, Y.: Tunesensor: A semantic-driven music recommendation service for digital photo albums. In: Proceedings of the 10th International Semantic Web Conference. ISWC2011 (October 2011)
2. Giuliano, C., Gliozzo, A.M., Strapparava, C.: Kernel methods for minimally supervised wsd. Computational Linguistics 35(4), 513–528 (2009)
3. Guerini, M., Strapparava, C., Stock, O.: Ecological evaluation of persuasive messages using google adwords. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics. ACL2012, vol. abs/1204.5369 (July 2012)
4. Hayes, C., Cunningham, P., Massa, P.: An on-line evaluation framework for recommender systems. Tech. Rep. TCD-CS-2002-19, Trinity College Dublin, Department of Computer Science (2002)
5. Jannach, D., Zanker, M., Felfernig, A., Friedrich, G.: Recommender Systems: An Introduction. Cambridge University Press (2011)
6. Lašek, I.: Dc proposal: Model for news filtering with named entities. In: Aroyo, L., Welty, C., Alani, H., Taylor, J., Bernstein, A., Kagal, L., Noy, N., Blomqvist, E. (eds.) The Semantic Web – ISWC 2011, Lecture Notes in Computer Science, vol. 7032, pp. 309–316. Springer Berlin / Heidelberg (2011)
7. Lv, Y., Moon, T., Kolari, P., Zheng, Z., Wang, X., Chang, Y.: Learning to model relatedness for news recommendation. In: Proceedings of the 20th international conference on World wide web. pp. 57–66. WWW '11, ACM, New York, NY, USA (2011)
8. Mason, W., Suri, S.: Conducting behavioral research on amazon's mechanical turk. Behavior Research Methods 44, 1–23 (2012)
9. Mendes, P.N., Jakob, M., García-Silva, A., Bizer, C.: Dbpedia spotlight: shedding light on the web of documents. In: Proceedings of the 7th International Conference on Semantic Systems. pp. 1–8. I-Semantics '11, ACM, New York, NY, USA (2011)
10. Negri, M., Bentivogli, L., Mehdad, Y., Giampiccolo, D., Marchetti, A.: Divide and conquer: crowdsourcing the creation of cross-lingual textual entailment corpora. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. pp. 670–679. EMNLP '11, Association for Computational Linguistics, Stroudsburg, PA, USA (2011)
11. Pazzani, M., Billsus, D.: Content-based recommendation systems. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) The Adaptive Web, Lecture Notes in Computer Science, vol. 4321, pp. 325–341. Springer Berlin / Heidelberg (2007)
12. Thalhammer, A., Ermilov, T., Nyberg, K., Santoso, A., Domingue, J.: Moviegoer - semantic social recommendations and personalized location-based offers. In: Proceedings of the 10th International Semantic Web Conference. ISWC2011 (October 2011)
13. Ziegler, C.N., Lausen, G., Schmidt-Thieme, L.: Taxonomy-driven computation of product recommendations. In: Proceedings of the thirteenth ACM international conference on Information and knowledge management. pp. 406–415. CIKM '04, ACM, New York, NY, USA (2004)