Semantic Information Filtering - Beyond Collaborative Filtering

Ivo Lašek
Faculty of Information Technology
Czech Technical University in Prague
Prague, Czech Republic
lasekivo@fit.cvut.cz

Peter Vojtáš
Faculty of Mathematics and Physics
Charles University in Prague
Prague, Czech Republic
vojtas@ksi.mff.cuni.cz

ABSTRACT

In this paper we introduce our idea of a semantic information filtering system. Contrary to traditional information filtering systems exploiting information retrieval techniques to select relevant data, we propose a workflow exploiting semantic information obtained from the web. Our system utilises the structured information crawled from the semantic web to improve the process of extracting the information from unstructured data sources. Moreover we suggest the ways of incorporating a user interaction in the whole process.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval; D.2.11 [**Software Engineering**]: Software Architectures

General Terms

Architecture, System

Keywords

Semantic Web, Information Filtering, Semantic Search

1. INTRODUCTION

With the growth of the Internet we are confronted with huge amounts of information everyday. In our daily lives we process more data than ever before. The information come from everywhere. The main sources include news, blogs or social networks. We have still more powerful tools to increase our productivity. But the information overflow, we are facing, poses a serious risk. It is crucial to develop new ways of information filtering which would effectively decrease the information overload.

One possible way is to exploit the advantages of semantic web technologies. We can try to identify real world entities in the texts and maintain some background knowledge about their properties. With this semantic information about the data, we can not only effectively suggest interesting topics and filter out the unimportant ones. We can also give the user the reason why we suggested the particular topic or the news article. This is usually not possible when using collaborative filtering, where the mechanism of recommendation remains hidden from the users and the user can not do much to influence it directly.

Another problem by collaborative filtering is, that we need to keep track of many users whose interests overlap. From this point of view the content based recommender systems are more flexible, while incorporating the news feeds from social networks for instance. There are often only few users reading the same message. Actually sometimes you may be the only person who reads it.

But when we talk about the content based recommending by news articles, we do not have many attributes which would serve us to check against a particular user profile. We can compare the articles based on the text properties using information retrieval techniques. But knowing at least some semantic information about the articles would help. In Section 6, we compared traditional keyword based information retrieval techniques with the semantic approach.

In this article, we address the problem of extracting semantic information from news articles and their further use for information filtering. We introduce a basic workflow to improve the results of named entity recognition with the help of a user feedback. Our workflow incorporates also the social aspects of user motivation to annotate information. The annotations then serve as a base for further improvements of automated information extraction. We propose the representation of a user profile for semantic information filtering.

The rest of the paper is organized as follows. First we describe our idea of the news filtering in section 2. Related work is recalled in Section 3. Section 4 introduces the proposed information filtering workflow, also the idea of maintaining a user profile based on the semantic information is introduced in section 4.5. In Section 5 we briefly describe the organization of the local storage of the system. Section 6 presents our experimental results. Finally, we conclude in Section 7.

2. NEWS FILTERING USE CASE

A large-scale human interaction study on a personal content information retrieval system, carried out by Microsoft [6], demonstrated that: "The most common query types in the logs were People / places / things, Computers / internet and Health / science. In the People / places / thing category, names were especially prevalent. Their importance is highlighted by the fact that 25% of the queries involved people's names... In contrast, general informational queries are less prevalent."

In our use case we focus at collecting data about people, organizations and places from news articles. We consider not only newspapers. Under news articles we understand also the posts of your favourite blogs, tweets of your friends or information from other social networks (e.g. news feeds on Facebook). In the articles, we then try to identify persons, organizations and places the particular article is writing about. We enrich the article with this extracted semantic information. Additionally, we register the time and the source of the information. We believe most information from news can be very well described precisely by the combination of the subject (a person or an organization), the time (when it happened) and the place (where it happened). If we have these information, we can effectively search and filter the news.

The users of the semantic information system can set up their profile indicating what they are interested in - selecting persons and organizations they want to read about. They can further indicate their favourite information sources. Optionally they can define a time period, they are interested in. The system will then deliver the information matching their criteria. It can deliver information from similar data sources added by other users with similar profiles too.

Finally the users are provided with the possibility to rate, how well the offered information fits their needs. The rating is then used to enhance their profiles - to precise their preferences. Also it is possible to mark any particular information (a word, a name or a sentence) which is interesting for them and annotate this information. The annotations of the same article from other users can be displayed.

This is important to motivate people to annotate the articles. One gain is the refinement of the own user profile to get more accurate results. The other motivation is the social aspect of sharing the information among users of the systems. You can display what other users marked as interesting in the same article you are reading.

3. RELATED WORK

As our workflow starts with automatic annotation of unstructured text coming from news articles, our inspiration comes mainly from two projects. SemTag [4] developed in IBM Almaden Research Center introduces the idea of automated semantic annotations. They show it is applicable in the large scale as well. SemTag was applied to a collection of 264 million web pages producing 434 million annotations.

The other project is KIM platform [10]. KIM platform focuses at the named entity recognition and the recognition of a limited amount of phrases. KIM platform further enables querying the extracted data and maintains interesting measures such as time lines. The time lines show, how often was a particular entity mentioned in a predefined period of time.

Many other systems are mentioned in the survey papers about web information extraction [1] and semantic annotation platforms [11]. The more recent survey provides an overview of Semantic APIs [5] that can be used for the named entity recognition too.

Our aim is not to create another semantic tagging or a named entity recognition system. We rather concentrate at using the listed existing solutions and adding our information filtering logic. Further we incorporate the feedback from the users in the form of their own annotations which are then incorporated in the whole knowledge base.

Both the mentioned projects SemTag and KIM platform use their own well defined knowledge bases. In the case of SemTag it is TAP [1] in case of KIM the data are collected from several structured databases. These data are clean, but tend to become obsolete in time and not ideally extensible. We believe the real challenge is in using live datasets crawled from the semantic web as a knowledge base.

In order to achieve this requirement the system would behave more like a semantic search engine. Specifically a search engine that works on a higher abstraction level and groups the data from more semantic documents under logical entities which correspond to real world objects. Otherwise there would not be possible to group the articles effectively by the particular entities. Thus for example the construction of time lines (that we mentioned by KIM platform) would be impossible. Representative examples of such a semantic search engines maintaining the entity centric approach are SWSE [9, 8] and Falcons [2].

4. SEMANTIC INFORMATION FILTERING WORKFLOW

In this section we introduce the general workflow of the whole system (see Figure 1).

4.1 Inputs

We consider three types of inputs.

- **Unstructured data:** In our use case the unstructured data is represented by web pages with news articles.
- **Semantic web entities:** When the users create their profiles, they need to indicate what they are interested in. They have to select some topics (persons, organizations and places in our use case) - we call them *semantic web entities*. To construct the knowledge base, we need to crawl the structured data from semantic web data sources. There are many well managed data sources we may obtain the information from. The data sources relevant for our use case include among others DBpedia[2], Freebase[3] or New York Times news vocabulary[4].

[1]http://ksl.stanford.edu/projects/TAP/
[2]http://www.dbpedia.org
[3]http://www.freebase.com
[4]http://data.nytimes.com/

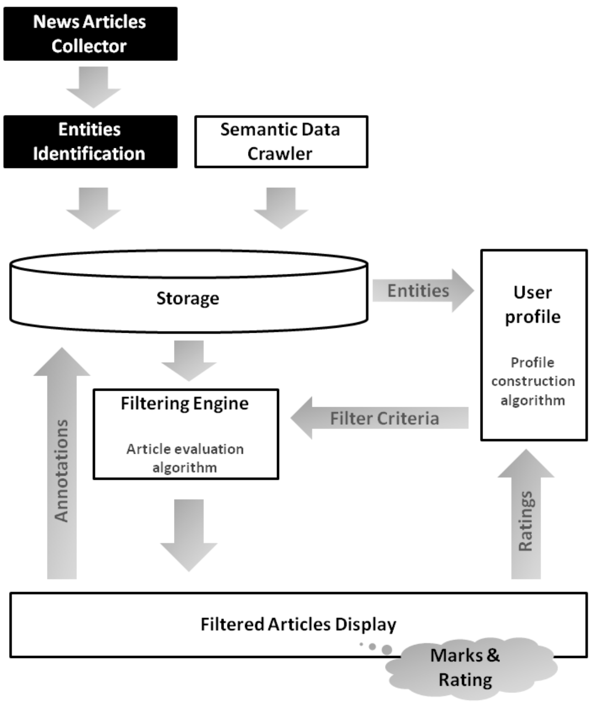

Figure 1: Semantic Information Filtering Workflow.

- **Annotations:** The annotations result from the feedback of the users of our system. Any time the user sees something important or interesting in any article, she or he can mark the information and indicate what that information means. The annotation tool is described in Section 4.4. The annotations then help us to extract more structured information from the articles.

4.2 Entities identification

Before storing the crawled article to our local repository, we identify the entities the article is writing about. We try to find and annotate the entities automatically. A basic approach to this task was introduced in the SWSE project [8]. The semantic search engine SWSE locates the description of a semantic entities in a plain text by searching for the values of their Inverse Functional Properties. This is more or less a brute force solution. Another more sophisticated way exploiting linguistic methods and inductive logical programming is introduced in [7] and [3].

Exploiting the linguistic analysis may precise the results of entities identification. We can work with the context of the occurrence of the entity. Consider the sentence: "This article is not writing about the XYZ company". If we do not understand the context, we would mark that article as if it was describing the entity XYZ company. But with linguistic analysis it is possible to identify the negative form.

There are also many ready made tools focusing at named entity recognition. For example OpenCalais[5], Zemanta[6], Epiphany[7] or KIM platform[8]. For our purposes the closest to our needs is KIM platform. It uses a semantic knowledge base and maps the articles to entities in this knowledge base. Searching the articles is then supported by the knowledge base and enables passing abstract queries like "find articles about companies established in 2010". The parts, where the workflow fully overlaps with the solution provided by KIM platform are denoted by black color in the Figure 1. Partially we adopt the ideas of indexing articles by entities and passing structured queries supported by the knowledge base.

Additionally, we extract the entities marked with the annotations coming from users, if there are any. We intend to use the user annotations to correct the process of automatic annotation and to improve its results. Potentially the automatic annotation system could learn from the user input.

4.3 Semantic Data Crawler

The creation of the semantic knowledge base is supported by a semantic web crawler. The crawler collects documents from predefined semantic web data sources and stores them locally in the semantic storage. The storage contains individual entities and information about them extracted from semantic web documents. We try to match the entities based on their URIs and the values of their Inverse Functional Properties. Our inspiration for this approach comes from [8]. However, matching of the entities is difficult. The URIs are not always set properly. Despite different URIs many sources describe the same thing.

Using Inverse Functional Properties is also error prone. They are often misused or contain incorrect values. For the future use, it is necessary to develop more sophisticated methods of matching the entities. Some ontology matching techniques may be incorporated. Also the quality criteria should be defined. Otherwise, the repository can easily become flooded by data from documents containing only few information about the entity to be matched with corresponding entity in the repository.

4.4 Filtered Articles Display

The filtered articles are presented to the end user in the form of a personalised news feed. The user can provide a feedback, to indicate his interests. The feedback has two forms:

- **Rating:** Users can rate the offered articles and indicate thus, how well the article fits their needs. There are only two degrees of the rating. Either the article is marked as interesting (the entities contained in the article are added to the profile) or it is marked as uninteresting (the contained entities are blacklisted).
- **Annotations:** The entities resulting from user annotations are added to the user profile. User can thus define interesting facts more precisely and not only at the level of the whole article.

[5]http://www.opencalais.com/
[6]http://www.zemanta.com/learn/
[7]http://projects.dfki.uni-kl.de/epiphany/
[8]http://www.ontotext.com/kim/

The annotation tool works as a plugin to a web browser. Any time a user sees something interesting, she or he can use the plugin to mark the information on a web page. The plugin then offers the possibility to indicate what that information means. This is done by searching the storage of the system for suitable entities. The user annotation is then sent back to the system and stored. Additionally the user profile is updated with the annotated entities to precise user preferences.

Users can thus view annotations created by other users and are motivated to create their own to get better results. With every created annotation the system gets better information about user preferences represented by their profiles (see section 4.5).

4.5 User profile

Users can select the entities, they are interested in. They can select the entities directly or by defining a structured query which is then performed against the semantic repository and the resulting set of entities is added to the user profile. The system then selects only articles that fit such criteria - write about the selected entities.

In a similar way, users can determine the set of entities they are not interested in and form a blacklist. Articles containing these entities are filtered out.

The user profile (after its first initialization) is maintained automatically, using a user feedback.

For calculation of the ratings we use a weight similar to the tf-idf weight used in information retrieval. Analogically we compute the ef-idef (entity frequency - inverse document entity frequency) as follows:

$$ef_{i,j} = \frac{e_{i,j}}{\sum_k e_{k,j}} \tag{1}$$

where $e_{i,j}$ is the number of occurrences of the considered entity in a particular document d_j and the denominator is the sum of the number of occurrences of all entities identified in the document.

The importance of the entity $idef_i$ is then obtained by dividing the total number of documents by the number of documents containing the entity, and then taking the logarithm of that quotient.

$$idef_i = \log \frac{|D|}{|\{d : e_i \in d\}|} \tag{2}$$

Algorithm 1 shows update of the user profile, after the user has rated one particular article ($article_j$).

Any time an article d_j is rated by the user, the contained entities e_i are extracted and the rating $r_{i,j}$ for each entity is stored. Additionally, we store the information whether the user rated a particular entity or the whole article.

Algorithm 1 User profile construction

$Entities \leftarrow extractEntitiesFrom(article_j)$
for all $entity_i \in Entities$ **do**
  if $isPositiveRatingOf(entity_i, article_j)$ **then**
    $r_{i,j} \leftarrow 1$
  else
    $r_{i,j} \leftarrow -1$
  end if
  if $entity_i \notin UserProfile$ **then**
    $addEntityToProfile(entity_i, UserProfile)$
  end if
  $appendRatingToUserProfile(r_{i,j}, UserProfile)$
end for

Each entity in the user profile is characterized by its significance. Equation 3 shows the calculation of the significance s_i^t.

$$s_i^t = \sum_j (r_{i,j} \times ef_{i,j}^t \times idef_i^t) \tag{3}$$

The significance s_i^t of individual entities contained in an article is always evaluated in the time t of comparison of this article to the user profile. The ratings $r_{i,j}$ come from all the articles containing entity i rated by the user.

It is necessary to evaluate the significance dynamically in order to take into account the current state of the corpus of all articles. By $ef_{i,j}^t$ and $idef_i^t$ we denote the values of these metrics in a particular point in time t as both can change in time.

The importance of entities $idef_i^t$ may change, because the count of all articles $|D|$ grows in time. The proportion of occurrences of an entity in an article $ef_{i,j}^t$ may also change. During the work of the systems users may any time annotate and thus identify more entities in the text.

In case that the user rates individual annotated entities and not the whole articles, same rules apply, but $ef_{i,j}^t$ is always equal to 1.

Algorithm 2 shows the filtering process. Every article is compared to the user profile based on the extracted entities.

The comparison of an article to the user profile is based on the computed significance s_i^t of individual entities. The significance s_i^t of the extracted entities e_i is summed. If the article rating ar is positive, the article is considered relevant for the user.

5. STORAGE

The storage contains collected entities in the semantic repository and all the crawled articles in a local cache.

The articles are stored in two copies. One is a snapshot of the original page. The other copy contains the snapshot with all the annotations we were able to discover during the entities identification process and all the annotations coming from the users. The preservation of the annotations directly

Algorithm 2 Article evaluation - comparison to the user profile

$Entities \leftarrow extractEntitiesFrom(article)$
$ar \leftarrow 0$
for all $e_i \in Entities$ **do**
  if $e_i \in UserProfile$ **then**
    $ar \leftarrow ar + s_i^t$
  end if
end for
$isArticleInteresting \leftarrow ar > 0$

in a web page holds not only the information about the content of the annotation, but also the information about its approximate location on the web page.

Additionally the annotated articles keep track of the time of its publication, time of its crawling and the source they were crawled from. Every annotated article contains a pointer to its original snapshot. The annotated articles are indexed by the entities they contain.

The entities are indexed by the content of their textual attributes (e.g. names, labels, descriptions) to enable their fulltext search.

6. EVALUATION

We performed several experiments to evaluate the impact of exploiting a semantic knowledge base in connection with information filtering.

In our use case scenario we assume a user (let's call him Michael) is interested in newly emerging technology startup companies. Exactly, Michael wants the system to deliver the news about the companies set up in 2009 or later. Probably he just wants to collect an inspiration for his own projects. Michael expects the system to filter out all the general talks like "How to start a company", "What should you do, to run a successful startup". Actually this kind of articles rather bothers him.

On the other hand he expects the system to deliver the articles about new emerging companies, how quick is their growth, what is their business model and also the information about acquisitions of these companies. An example of the potentially interesting article can be "ReadyForZero Launches Debt Management Platform To The Public". ReadyForZero is the name of the company set up in 2010.

6.1 Used Data

Having this scenario in mind we collected 150 random articles about technology companies from TechCrunch[9]. These articles were carefully read by a human and 32 articles potentially interesting for Michael were selected. As the knowledge base for this case, we used CrunchBase[10]. CrunchBase is the free database of technology companies, people, and investors that anyone can edit. The main advantage for our scenario is that CrunchBase is freely available also in a machine readable format for developers using its CrunchBase API.

[9]http://techcrunch.com/
[10]http://www.crunchbase.com/

6.2 The Course of Evaluation

Our aim is to compare traditional information retrieval techniques for information filtering with our method supported by a semantic knowledge base. We simulate the behaviour of the system right after the initialization of the user profile. By testing the information retrieval approach we suppose use of a system like Google Alerts [11]. Google Alerts enables you to define your own search query. Whenever some new document fitting the search criteria is discovered by Google, you get an alert by e-mail.

After the investigation of the texts of the 150 collected articles we identified the word "startup" (and its other forms "start-up", "startups" etc.) as the most significant keyword fitting Michael's needs. We could also use for example dates (Michael is interested in companies set up in 2009, 2010 and 2011). But since the dates are very often mentioned in the articles and only in few cases there is a connection to the founding date of a company, the incorporation of dates gave us very poor results and resulted in too many wrong suggestions.

Further we had to define the important area of each web page which we dealt with. Since TechCrunch focuses mainly at startup companies, the word startup occurs on almost any web page. In fact, from our data set counting 150 pages, 144 of them contained a word startup at least once. Startups were mentioned in advertisements, or links to other articles from the server. So we constrained the search criteria to include only two kinds of input. First only the body of the article text and in a second case the body together with the attached comments.

For the evaluation of the approach using the semantic knowledge base we set up the Michael's profile to collect companies that have the property "founded" (read from CrunchBase) set to the date after 1.1.2009. In our experiment we queried the CrunchBase to get all the companies set up after 1.1.2009. During the evaluation we performed the named entity extraction with the help of the names of the companies obtained from CrunchBase.

6.3 Results - Prepared User Profile

We evaluated three types of approaches:

- Semantic approach - using the semantic background knowledge in combination with the defined user profile.
- Traditional information retrieval approach applied to the article bodies - using the most significant keywords.
- Traditional information retrieval approach applied to article bodies and attached comments.

In Figure 2 the number of correctly suggested articles is shown (i.e. true positives). We put also the total number of correct suggestions identified by a human in the graph for reference. With the semantic approach, we were able to deliver 27 correct articles out of 32 possible. The semantic approach outperformed the keyword based information retrieval approach mainly in those articles, where there was

[11]http://www.google.com/alerts

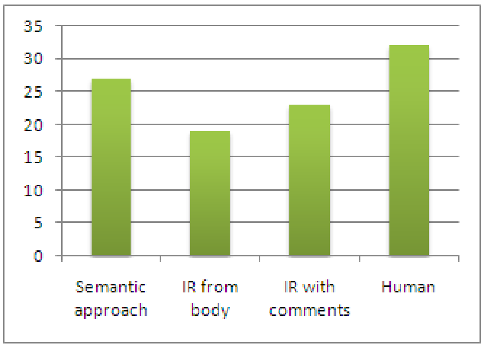

Figure 2: Number of correctly suggested articles (true positives) compared with the total number of interesting articles identified by a human.

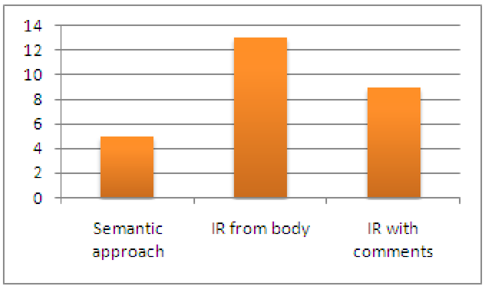

Figure 3: Number of articles which remained unidentified as interesting, although they were suggested by a human reader (false negatives).

just a description of the company products or activities, but it was not explicitly mentioned in the text that it is a newly started company. However, interesting is that this fact was often mentioned by the users in the comments. So the searching also in the comments brought better results counting 23 correct suggestions.

The Figure 3 shows a comparison of missed articles (i.e. false negatives). Practically all the misses in case of the semantic approach resulted from an incomplete knowledge base. There simply was no information about the founding date of the mentioned company. We did not find any case, where the article would describe a company but did not mention its name. Similarly the names of all the examined startups were quite distinctive and there was no problem with the confusion with the help of the knowledge base. In case of more general entities there would be certainly more errors resulting from wrong identification of entities. However our approach still counts with the possibility for the users to correct the particular errors in automatically generated annotations.

Finally we present the sum of wrong suggestions (i.e. false positives) in the Figure 4. That means those, which do not correspond to the articles selected manually. The semantic approach was wrong only in one case. It was in the article about a venture fund, where were enumerated the companies financed by the fund. The companies were correctly identified as startups, but the article was not about them. On the other hand the keyword based information retrieval

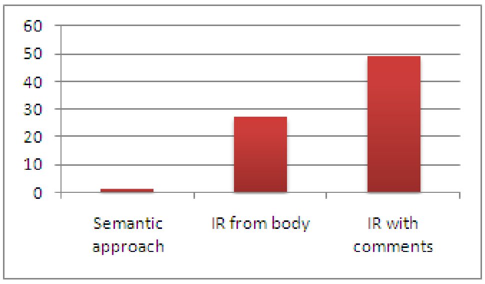

Figure 4: Number of incorrectly suggested articles. The articles which are probably not interesting for the user (false positives).

Table 1: Comparison of False Positives, True Positives, False Negatives, True Negatives, Precision and Recall by various approaches.

Method	FP	TP	FN	TN	Precision	Recall
Semantic approach	1	27	5	117	0.96	0.84
IR body	27	19	13	91	0.41	0.59
IR with comments	49	23	9	69	0.32	0.72

approach did not filter the articles very well. It got confused by statements like: "The company had more influence over our day to day lives than any other startup this decade." Which describes the company set up in 2004. Despite many occurrences of the word startup in the text, the article does not describe the company, which is a startup as well. This was the source of most mistakes. The situation gets even worse, when we dive into the comments from the users. The comments are full of statements like: "Nice chance for my startup..." under the article which describes some type of funding.

All the results are summarized in Table 6.3. From the results we can see that the semantic approach (ef-idef) delivers good results in terms of precision and recall. However, we have to state that the user profile was quite simple and focused at one particular use case. In real world scenarios the user profiles would be much more complex. Also the used knowledge base is well managed and only rarely the needed information was missing. This does not have to be true when we consider using data obtained from semantic web generally.

6.4 Results - Cold Start

Another issues arise when we consider a cold start, when a user does not define any query specifying entities he is interested in. We can still offer him random articles and try to learn his preferences. On the small data set, we were working with, the semantic approach failed.

In order to test the used techniques, we formed a training and a testing set. Each set contained 70 articles, 16 were correct and 54 were potentially uninteresting for the user.

We initialized the user profile with entities representing companies described in the articles from the training set. But because the sets of the entities identified in the testing and training set of articles did not overlap, the system was not able to identify any other article as interesting for the user.

On the other hand, the information retrieval approach gave more stable results. We used classical *tf-idf* metric with a set of stemmed words extracted from the body of the correct articles. Even on a small training data set of 16 correct articles, it was able to identify relevant keywords including: startup, launch and company.

The cold start problem arises whenever the user initializes his profile with only a small number of specific entities. The user is then informed only about these entities. Sometimes this is desirable. But often, we are interested in a broader concept rather than a small set of facts we selected manually. We identified two possible solutions:

- **Hybrid approach:** Especially in these situations, it may be beneficial to combine the semantic approach with collaborative filtering. We can propose not only articles containing exactly the selected entities, but also the articles containing entities from profiles of users with similar preferences.
- **Attribute analysis:** We can try to identify same values of some attributes of the entities in the user profile. If a significant amount of entities has same value of an attribute, we can deduce a broader concept. This concept can be used to identify another entities possibly interesting for the user. For example, if 30% of entities in our user profile have the value of the property *Founded* set to *2010*, we can try to find another entities with this property set to the same value and add them to the user profile. However, this is rather a naive approach and further evaluation needs to be done, in order to determine the significant number of occurrences of the attribute value in the user profile.

  This approach in its simple form is still not able to identify more complex concepts. The fact that all the companies were *Founded after 2009* remains undetected.

7. CONCLUSION

We introduced a semantic information filtering workflow and proposed a model to maintain a user profile based on the preferences according to semantic information extracted from news articles. We evaluated the semantic approach to information filtering on an example of news articles about technology companies.

When compared with collaborative filtering, the semantic approach enables the work of the system with even a single user. The semantic approach can also give the user a reason why a particular article was offered to him. The user can thus directly influence the filtering criteria.

We showed that this approach gives good results for properly set user profile. But there still remain unresolved issues regarding the cold start problem. One of possible solutions of the problem is to use the semantic approach in combination with collaborative filtering.

8. FUTURE WORK

In our future work we will focus at the construction of a comprehensive knowledge base with the help of data obtained from semantic web. We also intend to perform more extensive experiments evaluating the reliability of our user profile model.

We plan to develop a working platform based on the introduced principles, which would enable us to collect the feedback from real users. Thanks to the feedback from users we can develop several measures of the reliability of the system. First we can measure the explicit feedback - how often the users rate the filtered article positive (as interesting) and compare it with the count of negative ratings.

Additionally we can use the implicit feedback too. For example the time the users spend reading individual articles. If the user spends on the page more than a predefined time period (e.g. 30 seconds) we consider the article interesting. We can then compare the results of such a behaviour with the prediction about the article counted based on the user profile.

The results of the automatic entities identification and information extraction can be compared with the annotations created by individual users. This measure can serve us to evaluate the success of different extraction methods.

Acknowledgements. This work has been partially supported by the grant of The Czech Science Foundation (GAČR) P202/10/0761 and by the grant of Czech Technical University in Prague registration number OHK3-002/11.

9. REFERENCES

[1] C.-H. Chang, M. Kayed, M. R. Girgis, and K. Shaalan. A survey of web information extraction systems, 2006.

[2] G. Cheng, W. Ge, and Y. Qu. Falcons: searching and browsing entities on the semantic web. In *Proceeding of the 17th international conference on World Wide Web*, WWW '08, pages 1101–1102, New York, NY, USA, 2008. ACM.

[3] J. Dedek and P. Vojtas. Fuzzy classification of web reports with linguistic text mining. In *Web Intelligence and Intelligent Agent Technologies, 2009. WI-IAT '09. IEEE/WIC/ACM International Joint Conferences on*, volume 3, pages 167 –170, 2009.

[4] S. Dill, N. Eiron, D. Gibson, D. Gruhl, R. Guha, A. Jhingran, T. Kanungo, S. Rajagopalan, A. Tomkins, J. A. Tomlin, and J. Y. Zien. Semtag and seeker: bootstrapping the semantic web via automated semantic annotation. In *Proceedings of the 12th international conference on World Wide Web*, WWW '03, pages 178–186, New York, NY, USA, 2003. ACM.

[5] F. Dotsika. Semantic apis: Scaling up towards the semantic web. *International Journal of Information Management*, 30(4):335 – 342, 2010.

[6] S. Dumais, E. Cutrell, J. Cadiz, G. Jancke, R. Sarin, and D. C. Robbins. Stuff i've seen: A system for personal information retrieval and re-use. In *SIGIR '03*, pages 72–79. ACM Press, 2003.

[7] J. Dědek and P. Vojtáš. Linguistic extraction for

semantic annotation. In C. Badica, G. Mangioni, V. Carchiolo, and D. Burdescu, editors, *Intelligent Distributed Computing, Systems and Applications*, volume 162 of *Studies in Computational Intelligence*, pages 85–94. Springer Berlin / Heidelberg, 2008. 10.1007/978-3-540-85257-5_9.

[8] A. Harth, A. Hogan, R. Delbru, J. Umbrich, and S. Decker. Swse: answers before links. In *In Semantic Web Challenge*, 2007.

[9] A. Harth, A. Hogan, J. Umbrich, and S. Decker. Swse: Objects before documents!

[10] B. Popov, A. Kiryakov, D. Ognyanoff, D. Manov, and A. Kirilov. Kim - a semantic platform for information extraction and retrieval. *Nat. Lang. Eng.*, 10:375–392, September 2004.

[11] L. Reeve and H. Han. Survey of semantic annotation platforms. In *Proceedings of the 2005 ACM symposium on Applied computing*, SAC '05, pages 1634–1638, New York, NY, USA, 2005. ACM.