Term Ranking and Categorization for Ad-hoc Navigation

Ondrej Ševce, Jozef Tvarožek, and Mária Bieliková

Slovak University of Technology, Faculty of Informatics and Information Technologies,
Ilkovičova 3, 842 16 Bratislava, Slovakia
ondrej.sevce@gmail.com,[jtvarozek, bielik]@ fiit.stuba.sk

Abstract. Processing information in web pages and navigation on the web can take significant amount of time for users, requiring them to employ higher cognitive processes such as generalization and categorization. Providing users with annotated entities and terms contained in the text, and adaptive navigation based on these terms could help with the comprehension and better their orientation in the information space. In this paper, we present a method for adhoc navigation based on automatic terms retrieval, ranking and categorization. Recognized terms and categories are used as keywords for search in available content offering information spaces. Retrieved hyperlinks can be browsed by the user, while terms and categories gained from the last analyzed page are still available. Finally, the method includes user profiling, which enables grouping of the users based on their preferred terms and categories. Our results show that ad-hoc navigation can ease access to relevant related content on the web.

Keywords: term, category, navigation, conceptual user profile

1 Introduction and related work

Comprehension and interpretation of the text in web pages and navigation in the web information space take significant amount of time for many users (lost in hyperspace problem [10]).In particular, news articles typically contain various entities (persons, places, events), each having its own context that is easily recalled by humans by recollecting their previous personal experiences regarding these entities, posing a great challenge for machine processing. Systems for entity extraction from unstructured text are either domain specific, for example Essie which operates in medical domain [7], or domain independent, for example the user-friendly Wikify! System [8], which provides descriptions of entities gathered from Wikipedia, producing a "wikified" page to the user.

When extracting entities and terms, one of the issues to deal with is entity disambiguation. Entities and terms appear in the text in their "surface form", which may refer to various interpretations of the entity. This ambiguity can be eliminated by considering contextual evidence (words or other entities that

describe or co-occur with the entity) and category tags (which describe topics to which the entity belongs to) [4], or by machine learning on large data sample [9]. Category tags can be operationally retrieved from available folksonomies using graph algorithms, also providing the corresponding tag hierarchies [2]. The number of available entity extraction tools is increasing, and latest approaches tend to employ more than one extraction system, thereby increasing both entity recall (more systems recognize more entities) and precision [6].

When providing entity or term extraction results to a user, it may be valuable to assign relevance rating to entity, or to sort entities in order of relevance. Term rating and term ranking are tight together, as the higher rating of the term leads to its position closer to the top of the list. There are approaches that rank terms based on semantic techniques, like for example term expansion used along with terms and documents mapping into L2 space, and computing the inner product of this space to express similarity [14]. To adjust terms similarity, the sets of terms senses are compared. In [12] the term relevance scoring computation is based on considering term to document relations and also term to term relations. The method involves creation of indexed ontology, which provides valuable metadata for search refinement.

Search behavior of users shows that when navigating to the target, instead of using keywords, users navigate with small, local steps using their contextual knowledge as a guide [13]. Adaptive navigation that is based on retrieved entities and terms can thus have positive impact on user's sense of orientation in the web environment. There are several techniques which support adaptive navigation, such as annotation, sorting, hiding or generating of hyperlinks. Our approach is based on generating hyperlinks; in particular, it provides dynamic recommendation of relevant links [3].

In this paper, we propose an ad-hoc navigation method which relies on short-time user preferences. Terms and categories recognized in the text selected by the user are used as keywords for search in available content offering information spaces. Terms and categories are retrieved using shallow linguistic processing, which proved sufficient results for the purpose of keywords extraction. Based on keywords identified, the method provides links extracted from tweets and bookmarks retrieved from popular online systems Twitter and Delicious. The user can browse the web information space, while the context of the last analyzed page (represented by extracted concepts and recommended links) is still available. The ad-hoc navigation is engaged by explicit user's action, behaving as on-demand service.

Our approach frees users from devising relevant keywords, and gives them a stable context, which can be used as a basis in the web navigation. The difference between our proposed method and other existing methods for term extraction is in the ranking of retrieved terms, which in our case focuses the user's attention to the most important terms available in processed text. Another aspect of our method is that it is tightly integrated with the navigation. The emphasis was put on minimal user's effort simultaneously with providing wide range of navigation possibilities. We proceeded towards this goal also by integrating user interface to

the web browser, what enables easy access to the methods results. Further, our method includes user profiling which enables grouping of users based on their preferred terms and categories.

Following our method, we developed a system called *Marquess*, which includes web service capable of processing texts and returning machine-ranked terms and categories. It also supports user profiling based on principles of [11]. For the client side, a Mozilla Firefox add-on was developed, which enables communication with *Marquess* and other web services directly from browser window.

2 Method of ad-hoc navigation using term ranking

We propose a method for automatic term retrieval, ranking, categorization, and adhoc navigation, which employs also user profiling. In the following text, we use words *term, category*and *concept*, as follows: *term* – the surface form [4] oof abstract or concrete entity, as it occurs in the text (for example Barrack Obama); *category* – the surface form of some common characteristic or some generalization of related entities (for example Presidents of the United States); *concept* – term or category.

The typical use case of our method consists of the following steps (see figure 1):

1. Select a web page or part of its text and send it for processing.
2. Browse retrieved terms and categories.
3. Add / remove terms and categories to / from the user profile.
4. Navigate the user to other pages upon selected concept(s).

The user looks up interesting content on the current web page. She may choose to send the whole page or a part of it for term extraction. The set of terms and ccategories is then ordered by ranking, and the user is enabled to pick preferred conceptts into her profile, or select concept(s) and make it the basis for requesting navigatioonal paths from other information spaces. Concepts are selected explicitly by the useer, who is motivated by the need of additional information about the concept. Retrieved navigational links are provided to the user, and their target pages can becomme sources for subsequent term extraction. The process may be repeated multiple ttimes (see Figure 2) until the user is satisfied with results.

2.1 Retrieving terms

To retrieve terms, we employ shallow linguistic analysis in which we distinguish between original and nested occurrence of a term. For example, the text *„The White House Office of Health Reform said the process was going really well“* conttains terms *“White House Office of Health Reform”, “White House”,* and *“Health Reform”*. We distinguish two types of term occurrences:

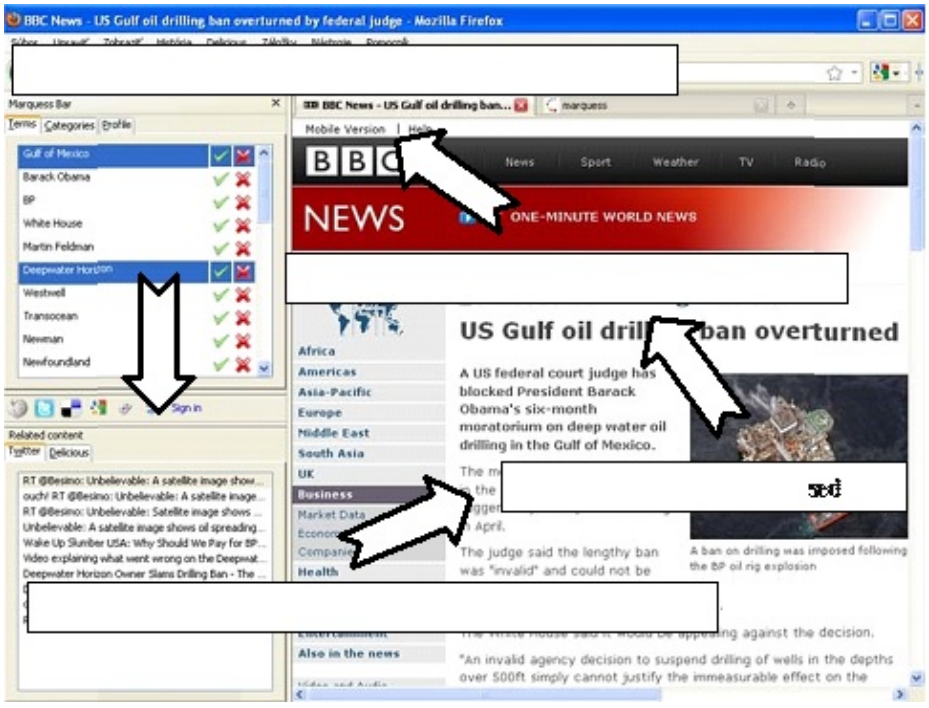

Fig. 1. Method use case – four steps, which enable concept based ad-hoc navigaation.

- *Original occurrence of a term* – the term in the text is not part of any other term and it logically fits into the context of text. In the stated example, it would be the term *"White House Office of Health Reform"*.
- *Nested occurrence of a term* – the term is part of other term, which ggives more detailed information and fits better into the context of text. In the stated eexample, it would be the terms *"White House"* and *"Health Reform"*.

Similar approach was used in [5] where author considers occurrence of nested "candidate" terms, which are included in "longer candidate" terms. In our approach, the machine term ranking requires counting of original and nested occurrences of terms in the text, for which we propose following algorithm: t – Vector of tokens (analyzed text), T – Text words count, N – Maximal length of term (wordss count), o – Occurrence vector of length T; each position contains index of term, in which the token occurred at last, c – Vector of retrieved terms, oo – Vector oof original occurrences of terms, no – Vector of nested occurrences of terms.

The algorithm for term retrieving and occurrence ranking is as follows:

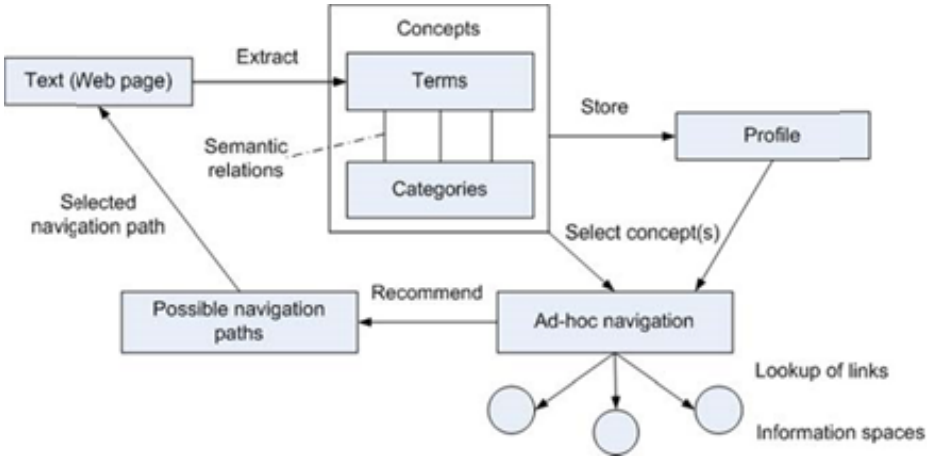

Fig. 2. Navigation recommendation cycle.

```
t = input_text.tokenize(), T = t.lenght o = vector(T),
oo = vector(T), no = vector(T) c = vector(0)

for (i = 1 to T) {   potential_term = t[i] + t[i+1] ...

t[i+N]   for (j = N to 0) {     if (j != N) removeLastToken
(potential_term)     if (isTerm(potential_term)) {
if (not c.contains(potential_term))
        c.addNewTerm(potential_term)       C = c.indexOf
(potential_term)       if (o[i] == -1 and o[i+j] == -1)
oo[C] += 1        else if (o[i] == o[i+j])
{ no[C] += 1; c[C].setNestedIn(c[o[i]]) }
for (k = i to i+j) o[k] = C     }  }  }
```

The searching for terms starts at the first token. The *potential term* is initialized to N subsequent tokens. If *potential term* is recognized as a new term it is added to the vector of retrieved terms. Next, the occurrence vector is checked (on positions of marginal tokens of current term), whether the actual term was already included in some other term. If not, the original occurrence of the actual term is increased, otherwise (if the actual term is enclosed in other term) the nested occurrence is increased and a relation between the terms is recorded. Next, the occurrence vector of every token included in the term is set to the index of the current term. In subsequent iterations, the last token of *potential term* is removed until only the first token is the *potential term*. Recognizing of terms continues in this way beginning with each token of the analyzed text.

We use the DBPedia dataset consisting of Wikipedia articles labels as the primary source of terms [1], i.e. as each term is directly related with article about

itself. The retrieving of terms is based on string matching. The DBPedia dataset consists of more than three million articles labels (in English version).

To enable real-time dataset search, we index the dataset's content using a hash map where key is the first word of article label, and value is the position of the first occurrence of this word in the dataset. Then, during the search for *potential term* in the dataset, the position of first word of *potential term* can be easily looked up in the hash map, and subsequently, the articles labels beginning with this word are compared with *potential term*. If a match is found the *potential term* is added to the list of retrieved terms.

2.2 Ranking terms and categories

The rating of a term is estimated by presented formula 1 (considering term occurrences and word count). Devising the weight coefficients presented in the formula is explained in the Section 3.

$$T = W.wc + W.oo + W.no \tag{1}$$

where Ti – relevance rating of term, wci – word count of term, ooi – original occurrences of term, noi – nested occurrences of term, Wwc – weight of the word count of term, Woo – weight of original occurrences of term, Wno – weight of nested occurrences of term.

Articles in Wikipedia are grouped in more than 400,000 categories. DBPedia offers the dataset of categories and relations between articles and categories currently containing about two million records. We take advantage of the human-made relations when looking up categories of a particular term. Each category gains relevance rating based on the ratings of its related terms (the rating of category is the sum of ratings of related terms, see Formula 2). Categories are presented to the user in a separate list, ranked by machine-computed relevance, and simultaneously providing the user with more general information related to the analyzed text.

$$C = \sum_{j=1}^{n} Tj \tag{2}$$

where C – relevance rating of category, n – number of terms which occur in the text, and are related to the category, Tj – rating of term.

2.3 Ad-hoc navigation

Based on client's interactions with Marquess, information spaces are searched for additional content by using their online interfaces. The ad-hoc navigation is affected by user's choice of a page to be analyzed, and subsequently, by picking the concept(s) to be looked up in other information spaces. User is enabled to pick one or more concepts at the time. When multiple concepts are chosen the search string is built using these concepts in order of their ranking.

Client side of the implemented system is interacting with popular microblogging system Twitter, bookmarking system Delicious, provides simple Google search feature, and direct links to Wikipedia articles (for terms and categories). Twitter and Delicious services return tweets and bookmarks ordered by creation time (recent results appear first); we present them to users in this same order. The system extracts links from tweets and enables user to read the tweet, view target page of the link or home page of person who published the tweet. Opening the links included in tweets can be a direct way to obtain recent news and information. Delicious bookmarks usually provide more time insensitive content, guiding users to general information about the selected concept(s).

2.4 Profiling

In the proposed method, profiling partly depends on user's interactions with the system. After collecting and ranking of terms and categories, these are presented to the user in temporary lists. The user may interact with these lists by adding or removing terms and categories to or from her profile. When accumulated in profile, concepts create a base of user's preferences, and have universal usage, such as keywords for search engine queries, keywords for searching in folksonomies, matching RSS feeds, and so on. The profile contains also data about user's interactions with the system (as proposed in [11]) and relations between terms and analyzed documents, what makes it the base for further content recommendation.

3 Evaluation

To evaluate the proposed method, we performed three experiments. The goal of the first experiment was to set the weights coefficients of the formula 1 so that we would gain satisfactory term ranking. Second experiment evaluated weights coefficients gained in the first experiment by comparing human and machine rating of categories relevance. In the third experiment, we evaluated the relevance of links provided by the ad-hoc navigation.

In the first experiment, we evaluated different combinations of weights presented in formula 1. The machine rating of terms is based on original and nested occurrences of the term and on word count of the terms *(oo, no, wc)*. These parameters are multiplied by weights coefficients and their sum is the machine rating off the term (which implies the ranking of the term). To set the values of weights coefficients, we involved human experts in the experiment, and used their relevance rating too optimize weights values. Figure 33 shows the dependence of MSE on weights combinations (explained further).

Seven human experts rated relevance of terms found in six various articles from BBC news on the scale from zero to ten. Average human rating of each term was considered as the real relevance of the term. The system was optimized by testing combinations of weights (1,331 combinations, each weight Woo, Wno, Wwc was represented 11 times, on the scale from 0 to 10) to gain ratings closest

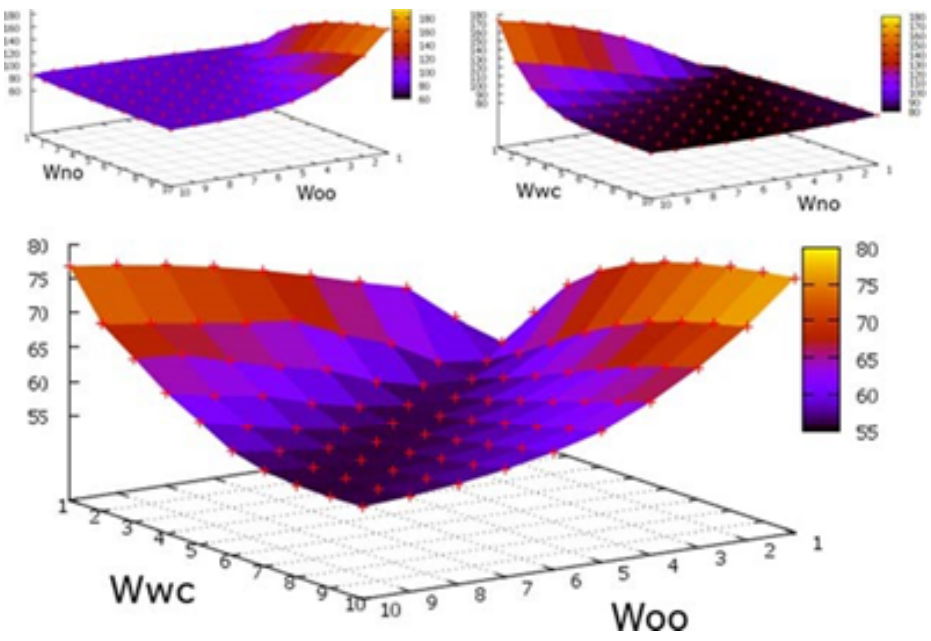

Fig. 3. Values of MSE computed from combinations of Wwc, Wno and Woo. Lower MSE means smaller difference between human and machine terms rating. The graphs show tthat certain combinations of weight coefficients Wwc and Woo gives results closest to the human rating.

to the average human ratings. The weight scale measure was set to one, as more fine scale measurement didn't significantly affect the results.

The quality of machine rating was evaluated by MSE (MSE was calculated for each article as the sum of the square roots of differences between average human rating of term and the machine rating of term, for each term retrieved from the article). The z-dimension of graphs in Figure 3 shows computed MSE for given weights coefficients combination. The coefficients combination that gave closest rating to the average human rating was Woo = 3, Wno= 4, Wwc = 2, MSE = 45.894.

In the second experiment, we evaluated optimized weights coefficients. We let the same group of human experts rate the relevance of categories related to retrieved terms. The relevance of category was marked by the expert by saying "yes, it is relevant to the text", or "no, it is not relevant to the text". Each category was rated by seven human experts, and the relevance of category was proportional to the number of "yes, it is relevant to the text" choices.

Totally, average MSE for one category rating was 7.18, which means that the average difference between human and machine rating was 2.68 on the scale from 0 to 10.

Table 1. Difference between normalized human and machine rating of categories related to the terms extracted from analyzed documents.

Article	Average rating MSE	Categories	Standard deviation of MSE
1	7.610	15.0	7.963
2	4.391	26.0	4.173
3	5.955	22.0	11.061
4	2.563	5.0	1.674
5	14.936	10.0	11.979
6	9.732	15.0	12.089
Average	7.531	15.5	8.156

In the third experiment, we evaluated the relevance of links provided by the ad-hoc navigation. The navigation links were looked up by selecting one, two or three terms with highest ranking. We evaluated total number of navigation links retrieved, number of relevant navigation links and finally number of off topic navigation links.

Table 2. Evaluation of ad-hoc navigation.

	Twitter			Delicious		
	1 term	2 terms	3 terms	1 term	2 terms	3 terms
Avg. keyw. count	1.71	2.57	3.85	1.71	2.57	3.85
Total tw./bkm.	8.86	11.43	8.29	12.86	7.53	2.64
Relevant to article	4.71	5.67	3.52	5.71	3.25	2.14
Relevant to terms	4.71	5.86	3.95	10.14	5.48	2.32
Off topic	1.14	0.29	0.00	0.68	0.30	0.11

Obtained results suggest the following findings about ad-hoc navigation: (i) Twitter reacts very fast on news articles, as the relevance of discovered links is nearly identical for terms and for the article (terms "gain" the relevance from being stated in related articles); (ii) Number of off topic navigation links is decreasing while the number of used terms increases (both Twitter and Delicious); (iii) Searching two highest-ranked terms brought the highest number of navigation links (total, relevant to article, relevant to terms) from Twitter; in the case of Delicious, using only the single highest-ranked term brought the most navigation links.

4 Summary and Discussion

In this paper, we have presented and evaluated a method for automatic terms retrieval, ranking, categorization and ad-hoc navigation. One of the key aspects of our method – term ranking is based on shallow linguistic analysis which appears to be sufficient for the purpose of ad-hoc navigation. When rating terms, three

weight coefficients are used (for each measured parameter). These weight coefficients were optimized by adapting machine rating of terms to average human relevance rating of terms.

Selecting and evaluating various combinations of weights (where one of weights was always set to zero) demonstrated that the importance of original occurrences and word count of term is higher than the importance of nested occurrences, although the best result was achieved while nested occurrences weight was higher than the other two weights. When rating categories with optimized weight coefficients, we gained quite unbalanced variation between human and machine ratings, although categories were related to the text via semantic relations with retrieved terms. By using humanmade semantic relations, every category offered to a human expert for rating was relevant to the text. The unresolved question is, if the experts knew about the relation of this category to the article; the relevance rating of categories may be influenced by this "nescience" of human experts. However, user's interest in categories and their related content is partially based on personal preferences, so the precise relevance order of categories may not be the crucial issue of the proposed method.

The ad-hoc navigation showed both strengths and weaknesses. The positive aspects of this kind of navigation are its context dependency and adaptability. These emerge from its integration with dynamic folksonomies providing data processed by collective intelligence. These data are not perfect, for example Twitter contains high amount of noise and redundancy. The noise is represented by off topic tweets reflecting current events or terms related to these events. Noise can be partly eliminated by filtering tweets without links.

The redundancy is difficult to discover, and thus not easy to eliminate. By frequent use of link proxies in tweets, many links in fact point to the same article. On the other hand, links retrieved from Delicious were affected by less noise and contained less redundancy. Delicious links also proved higher relevance when requesting bookmarks for articles regarding more significant or long-term topics, while some topics were ignored by Delicious users. These results confirmed that Twitter is a good resource of links for current topics, while Delicious provides links with more long-term usability. Therefore our ad-hoc navigation method could be useful for users demanding various information about actual events, or users with deeper interests in particular topics.

In the future, we plan to improve the navigation method and users profiling by discovering of similarities between users' profiles. Information stored in profiles should be used for relating users via compliant terms, categories and documents.

These relations may allow for a more sophisticated content recommendation. There are also unresolved issues about terms ambiguity, for example names of persons often refer to different persons. These issues should be eliminated by integrating additional services and gathering meta-data about terms retrieved from the texts.

Acknowledgments. This work was supported by the Scientific Grant Agency of SR, grant No. VG1/0508/09, the Cultural and Educational Grant Agency

of SR, grant No. 028-025STU-4/2010, and it is a partial result of the Research and Development Operational Program for the project Support of Center of Excellence for Smart Technologies, Systems and Services II, ITMS 25240120029, co-funded by ERDF.

References

1. Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. Dbpedia: A nucleus for a web of open data. In *The semantic web*, pages 722–735. Springer, 2007.
2. Michal Barla and Mária Bieliková. On deriving tagsonomies: Keyword relations coming from crowd. In *International Conference on Computational Collective Intelligence*, pages 309–320. Springer, 2009.
3. Peter Brusilovsky and Mark T Maybury. From adaptive hypermedia to the adaptive web. *Communications of the ACM*, 45(5):30–33, 2002.
4. Silviu Cucerzan. Large-scale named entity disambiguation based on wikipedia data. In *EMNLP-CoNLL*, volume 7, pages 708–716, 2007.
5. Ismail Fahmi. C-value method for multi-word term extraction. In *seminar in Statistics and Methodology*, 2005.
6. Francisco Iacobelli, Larry Birnbaum, and Kristian J Hammond. Tell me more, not just more of the same. In *Proceedings of the 15th international conference on Intelligent user interfaces*, pages 81–90. ACM, 2010.
7. Nicholas C Ide, Russell F Loane, and Dina Demner-Fushman. Essie: a concept-based search engine for structured biomedical text. *Journal of the American Medical Informatics Association*, 14(3):253–263, 2007.
8. Rada Mihalcea and Andras Csomai. Wikify!: linking documents to encyclopedic knowledge. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 233–242. ACM, 2007.
9. David Milne and Ian H Witten. Learning to link with wikipedia. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 509–518. ACM, 2008.
10. Malcolm Otter and Hilary Johnson. Lost in hyperspace: metrics and mental models. *Interacting with computers*, 13(1):1–40, 2000.
11. Michael J Pazzani and Daniel Billsus. Content-based recommendation systems. In *The adaptive web*, pages 325–341. Springer, 2007.
12. M Šimko and M Bieliková. Improving search results with lightweight semantic search. In *Proc. of the Workshop on Semantic Search, SemSearch*, pages 53–54, 2009.
13. Jaime Teevan, Christine Alvarado, Mark S Ackerman, and David R Karger. The perfect search engine is not enough: a study of orienteering behavior in directed search. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 415–422. ACM, 2004.
14. Peter Wittek, Sándor Darányi, and Chew Lim Tan. Improving text classification by a sense spectrum approach to term expansion. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, pages 183–191. Association for Computational Linguistics, 2009.