A Keyphrase-Based Paper Recommender System

Felice Ferrara, Nirmala Pudota, and Carlo Tasso

Department of Mathematics and Computer Science, University of Udine
Via delle Scienze, 206 – 33100 Udine, Italy
{felice.ferrara,nirmala.pudota,carlo.tasso}@uniud.it

Abstract. Current digital libraries suffer from the information overload problem which prevents an effective access to knowledge. This is particularly true for scientific digital libraries where a growing amount of scientific articles can be explored by users with different needs, backgrounds, and interests. Recommender systems can tackle this limitation by filtering resources according to specific user needs. This paper introduces a content-based recommendation approach for enhancing the access to scientific digital libraries where a keyphrase extraction module is used to produce a rich description of both content of papers and user interests.

Keywords: Recommender systems, content-based, keyphrase extraction, adaptive, personalization.

1 Introduction

Information access involves three main actors: the users (who are interested in obtaining knowledge), the document collections (which contain available information), and the access functions (which support the user in extracting knowledge from available resources) [1].

Web growth and evolution have changed the characteristics of both users and document collections. In fact, the participative Web allows a growing number of users to access and populate document collections in a simple way, producing larger and larger collections. As a result, document collections can be explored by a very large set of users, who access the repositories in order to satisfy various personal information needs. Unfortunately, this growing size of the digital information space prevents an effective access to knowledge due to the well-known phenomenon of information overload. Therefore, innovative ways to access Web contents are required. A viable solution to these problems is constituted by adaptive personalization, i.e. to first identify and model the specific information needs of the user (building in such a way a user profile) and to subsequently filter Web resources according to the individual user profile. Also scientific digital libraries (such as CiteULike[1], Elsevier[2] and PubMed[3]) show the above criticalities, and

[1] http://www.citeulike.org/
[2] http://www.elsevier.com/
[3] http://www.ncbi.nlm.nih.gov/pubmed

M. Agosti et al. (Eds.): IRCDL 2011, CCIS 249, pp. 14–25, 2011.

to avoid them, they have experimented various innovative mechanisms, such as email alert systems, RSS feeds, and recommender systems. The goal of recommender systems is to model user interests and to filter resources according to the identified user needs and interests.

The aim of this paper is to propose a content-based recommender approach for scientific digital libraries, which extracts keyphrases from papers in order to have a rich description of both resources and user interests. A keyphrase is a short phrase (typically constituted by one to three words) that provides a key idea of a document. A keyphrase list is a short list of keyphrases that reflects the content of a single document, capturing the main topics discussed and providing a brief summary of its content. In this work, the keyphrase lists extracted from papers which are relevant to a specific user are exploited to create his/her user profile. Then, in order to compute the relevance of a new article, the user profile is compared with the keyphrase list extracted from that article.

This paper is organized as follows: Section 2 introduces related works, focusing specifically on the keyphrase extraction task and on recommender systems for digital libraries. The domain-independent keyphrase extraction technique used to model the content of a resource and to produce the user profile is described in detail in Section 3, as well as the proposed recommendation approach. The evaluation is provided in Section 4, while future work and final considerations conclude the paper in Section 5.

2 Related Work

This section provides the reader with background concepts of keyphrase extraction (2.1) and recommender systems (2.2).

2.1 Keyphrase Extraction

Keyphrase extraction methods have been used successfully in *Information Retrieval (IR)* and *Natural Language Processing (NLP)* tasks, such as document indexing [2], classification [3], and automatic tagging [4].

Keyphrase extraction methods usually work in two stages: (i) a *candidate identification* stage, identifies all possible phrases from the document and (ii) a *selection* stage selects only few candidate phrases as keyphrases. Existing methods for keyphrase extraction can be divided into supervised and unsupervised approaches.

A *supervised approach* builds a model by using training documents that have already keyphrases previously assigned to them by humans. This model is trained to learn features of the relevant keyphrases (the keyphrases assigned by humans to the training documents) and then it is exploited in order to select keyphrases from unseen documents. *KEA* [5] is a notable supervised approach which uses a Bayesian classifier. KEA analyzes training documents by taking into account orthographic boundaries (such as punctuation marks, newlines, etc.) in order to find candidate phrases. In KEA two specific features are exploited as metrics in

order to rank candidate keyphrases: tf×idf (term frequency × inverse document frequency) and the position of the first occurrence of the keyphrase. Hulth [6] introduces linguistic knowledge (i.e., *POS, Part-Of-Speech tags*) in determining candidate sets: 56 potential *Pos-patterns* are used for identifying candidate phrases in the text. The experimentation carried out by Hulth has shown that, using a *POS tag* as a feature in candidate selection, a significant improvement of the keyphrase extraction results can be achieved. Another system that relies on linguistic features is LAKE (Learning Algorithm for Keyphrase Extraction) [7]: it exploits linguistic knowledge for candidate identification and it applies a Naïve Bayes classifier in the final keyphrase selection. All the above systems need training data (in a larger or smaller extent) in order to construct an extraction system. However, acquiring training data with known (i.e., already assigned) keyphrases is not always feasible and human assignment is time-consuming. Furthermore, a model that is trained on a specific domain, does not always produce adequate classification results in other domains.

The *unsupervised approach* eliminates the need of training data. It selects a general set of candidate phrases from the given document, and it uses some ranking strategy to select the most important candidates as keyphrases for the document. Barker and Cornacchia [8] extract noun phrases from a document and ranks them by using simple heuristics, based on their length, frequency, and the frequency of their head noun. In [9], Bracewell et al. extract noun phrases from a document, and then cluster the terms which share the same noun term. The clusters are ranked based on term and noun phrase frequencies. Finally, the top-n ranked clusters are selected as keyphrases for the document. The authors of [10] and [11] proposed unsupervised approaches based on a graph representation of documents. Such approaches use ranking strategies (similar to the PageRank algorithm [12]) to assign scores to each term. Keyphrase extraction systems that are developed by following unsupervised approaches are in general domain independent since they are not constrained by specific training documents.

2.2 Recommender Systems

Information overload is the main motivation for recommender systems: they support users during their interaction with large information spaces, directing them toward the specific information they need [13]. Recommender systems filter relevant content according to individual information needs of a specific user (in this paper, referred also as *active user*). In order to reach their aim, recommender systems can possibly exploit a suitable representation of user interests, goals, knowledge, and tastes, by monitoring and modeling implicit and/or explicit feedback provided by the user. By analyzing such user feedback, a recommender system is capable of computing a personalized rank for the set of available resources.

The most common classification of recommender systems takes into account the algorithm used to produce recommendations and identifies three classes of recommender systems: collaborative filtering, content-based, and hybrid recommender systems [14]. Collaborative filtering recommender systems filter resources

by using opinions provided by other people. Content-based recommender systems analyze the past user activities looking for resources she liked; they model resources by extracting some features (for example, topics or relevant concepts) from documents. The user profile is then built by identifying features which are interesting for the user. The relevance of a new resource for a user is then computed by comparing the representation of the resource to the user profile. Hybrid recommender systems variously combine results returned by collaborative and content-based recommender systems [15].

Recommender systems have been widely proposed to face information overload in digital libraries. The authors of [16] propose a collaborative filtering recommender system aimed at taking into account that each user of a digital library may be interested in several distinct topics. Their work focuses on innovative digital libraries which include Web 2.0 features such as social tagging: the active participation of Web 2.0 users is exploited in order to identify different topics of interest (ToI) of the users. This is achieved by clustering the tags utilized by a user, joining together tags with similar meanings. The similarity depends on the number of times two tags have been applied on the same resource. Such tag clusters allow to split resources tagged by the user into different collections, each one associated to a specific ToI. Only opinions of users interested into a specific ToI are then considered to compute recommendations. In particular, resources labeled by tags which are evaluated as more similar to the tags associated to a ToI are considered more relevant than other resources, and resources bookmarked by users more similar to the active user are more relevant than others as well.

A content-based approach has been proposed in [17] where authors and papers are modeled by trees of concepts: using the ACM Computing Classification System (CCS), the authors trained a vector space classifier in order to associate concepts of the CCS classifications to documents. The hierarchical organization of the CCS allows the system to represent user interests and documents by trees of concepts. A user profile and a paper representation are then compared by a tree edit-distance which computes a similarity measure among trees. The authors of [18] proposed a paper recommender system which takes into account authors' publications and the papers they cite in order to define a user profile and then compute recommendations. Term frequency is used to create a weighted vector of terms to model user interests: the textual content of both the papers written by the active user and papers cited by the active user are considered for building his profile. Similarly, resources are represented as vectors using the tf×idf metric. Then, the relevance of a document with respect to the active user depends on the cosine similarity among the user profile and the resource representation. PaperRank [19] is a PageRank-like method [12] to filter relevant papers by exploring the citation graph according to a seed of input documents. The approach is a modified version of the PageRank algorithm which gives higher relevance to the papers relevant for the active user.

3 The Proposed Approach

This section presents the proposed approach, whose general organization is shown in Figure 1. Our proposal is included in the Pirates project [20] aimed

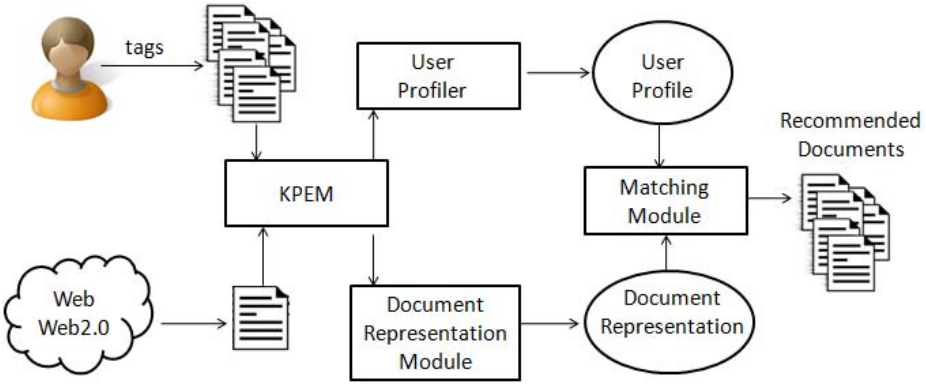

Fig. 1. General organization of the proposed approach

at studying and developing new advanced tools for Web information access, classification, retrieval, and extraction. One of the goals of the overall Pirates framework is to recommend, by means of various techniques, new relevant content (as well as classification information, such as tags [16][21]) in an adaptive personalized way. The part of the Pirates framework that we are presenting in this paper takes into account that a user has normally associated to him a set of documents of interest, which have been usually (but not necessarily nor exclusively) identified by tagging. So if a user has a set of tagged documents, these can be exploited by the KPEM (Key Phrase Extraction Module) for building his user profile. On the other hand, if a new document is extracted from the Web (for example by the IFT [22] content based filtering module present in Pirates), the same processing exploiting KPEM can be utilized for building a representation of the document. The Matching Module takes finally both the user profile and the document representation in order to compute a matching score, and to identify documents to be suggested to the user.

3.1 Extracting Keyphrases

Given a paper p, the unsupervised keyphrase extraction approach used in our proposal exploits the following three main steps: (i) extract candidate phrases from p (ii) calculate feature values for candidates (iii) compute a score for each candidate phrase from its feature values and filter the top weighted keyphrases. The following subsections illustrate the three steps.

Step1: Candidate Phrase Extraction. In this phase three main steps are exploited:

- ***POS tagging and n-gram extraction***. We assign a POS tag (noun, adjective, verb, etc.) to each token in the text (for English documents we use the Stanford log-linear part-of-speech tagger[4] while, for documents written in Italian, we developed ad-hoc an Italian POS tagger). Subsequently, n-grams are extracted and since keyphrases constituted by more than 3 words are usually very rare, we extract only all possible subsequences of phrases up to 3 words (uni-grams, bi-grams, and tri-grams).
- ***Stemming and stopword removing***. From the extracted n-grams, we remove all phrases[5] that start and/or end with a stopword. Partial stemming (i.e., unifying the plural forms and singular forms which refer to the same meaning) is performed using the first step of Porter stemmer algorithm [23]. To reduce the size of the candidate phrase set, we filter out some of them by means of a rule-based engine which uses POS tagging information: for example, uni-grams that are not labeled as noun, adjective, or verb are filtered, bi-grams and tri-grams are filtered according to [24], and so on.
- ***Separating n-gram lists***. Generally, in a document, uni-grams are more frequent than bi-grams, and bi-grams are more frequent than tri-grams. This introduces an undesidered bias in the subsequent computation of phrase features. In order to avoid this, we separate n-grams of different lengths and arrange them in three different lists, then treated separately.

Step2: Feature Computation. This step characterizes each candidate phrase (included in the three n-gram lists) by statistical and linguistic properties. The following five features are computed for each candidate phrase:

- ***Phrase Frequency***. It is the classical term frequency (tf) metric. But instead of calculating it with respect to the whole length of the document, we compute it with respect to each n-gram list. With reference to each n-gram list, the phrase frequency for phrase P in list L is:

$$frequency(P, L) = \frac{freq(P, L)}{size(L)},$$

  where $freq(P, L)$ is the number of times P occurs in L and $size(L)$ is the total number of phrases included in L.
- ***POS Value***. As described in [8], most author-assigned keyphrases for a document turn out to be noun phrases. For this reason, in our approach, we stress the presence of a noun in a candidate phrase while computing a POS Value for the phrase. A POS Value is assigned to each phrase by calculating the number of nouns (singular or plural) normalizing it by the total number of terms in the phrase. All remaining phrases which do not include at least one noun term are assigned a lower POS Value.

[4] `http://nlp.stanford.edu/software/tagger.shtml`.
[5] In our use of this term, we mean any n-gram (n=1,2,3) phrase.

- ***Phrase Depth***. This feature reflects the assumption/belief that important phrases often appear in the initial part of the document especially in news, articles, and scientific publications (e.g., *abstract, introduction*). We compute the position in the document where the phrase first appears. The Phrase Depth value for phrase P in a document D is:

$$depth(P, D) = 1 - [\frac{first\_index(P)}{size(D)}],$$

  where $first\_index(P)$ is the number of words preceding the phrase's first appearance and $size(D)$ is the total number of words in D. The result is a number between 0 and 1. Highest values represent the presence of a phrase at the very beginning of the document.
- ***Phrase Last Occurrence***. We give also importance to phrases that appear at the end of the document, as it is the case in scientific articles (i.e., in conclusion and discussion parts). The last occurrence value of a phrase is calculated as the number of words preceding the last occurrence of the phrase normalized to the total number of words in the document. The last occurrence value for phrase P in a document D is:

$$last\_occurrence(P, D) = \frac{last\_index(P)}{size(D)},$$

  where $last\_index(P)$ is the number of words preceding the phrase's last appearance and $size(D)$ is the total number of words in D.
- ***Phrase Lifespan***: the span value of a phrase depends on the portion of the text that is covered by the phrase. The covered portion of the text is the distance between the first occurrence position and last occurrence position of the phrase in the document. The lifespan value is computed by calculating the difference between the *Phrase Last Occurrence* and the *phrase first occurrence*. The lifespan value for phrase P in a document D is:

$$lifespan(P, D) = \frac{[last\_index(P) - first\_index(P)]}{size(D)},$$

  where $last\_index(P)$ is the number of words preceding the phrase's last appearance and $first\_index(P)$ is the number of words preceding the phrase's first appearance; $size(D)$ is the total number of words in D. The result is a number between 0 and 1.

As a result of step 2, we get a feature vector for each candidate phrase in the three n-gram lists.

Step3: Scoring and Ranking. In this step a score is assigned to each candidate phrase which is later exploited for the selection of the most appropriate phrases as representatives of the content of a document. The score of each candidate phrase is calculated as a linear combination of the 5 above features. We call

the resulting value the *keyphraseness* of the candidate phrase. The keyphraseness of a phrase P with a non empty feature set {f_1,f_2,...,f_5}, with non-negative weights {w_1,w_2,..,w_5} is:

$$keyphraseness(P) = \frac{\sum_{i=1}^{5} w_i f_i}{\sum_{i=1}^{5} w_i}$$

In the initial stage of the research, we have assigned equal weights to all features, yielding to the computation of the average. Therefore:

$$keyphraseness(P) = \frac{1}{n}\sum_{i=1}^{n} f_i,$$

where n is the total number of features, f_1 is the Phrase Frequency, f_2 is the Phrase Depth, f_3 is the phrase POS Value, f_4 is the Phrase Last Occurrence, and f_5 is the Phrase Lifespan.

Producing Final Keyphrases. The scoring process produces the three ranked lists $UniGrams(p)$, $BiGrams(p)$, and $TriGrams(p)$, each one containing respectively all the uni-grams, bi-grams, and tri-grams extracted from the paper p with their keyphraseness values. In order to filter only the most relevant keyphrases, the three output lists are pruned by removing keyphrases with a keyphraseness lower than a fixed threshold.

3.2 Computing Recommendations

Given a user, the papers that he tagged are considered relevant papers for building the User Profile. The profile is constituted by three (ordered) lists of weighted and stemmed keyphrases: the list of uni-grams (the uni-gram profile), the list of bi-grams (the bi-gram profile), and the list of tri-grams (the tri-gram profile).

More specifically, given the list of relevant papers $\{p_1, \ldots, p_n\}$ for the active user, we exploit the keyphrase extraction approach described above to extract uni-grams, bi-grams, and tri-grams from each relevant paper separately. This step produces three lists of weighted keyphrases:

1. the list of the weighted uni-grams $\{UniGrams(p_1), \ldots, UniGrams(p_n)\}$;
2. the list of the weighted bi-grams $\{BiGrams(p_1), \ldots, BiGrams(p_n)\}$;
3. the list of the weighted tri-grams $\{TriGrams(p_1), \ldots, TriGrams(p_n)\}$.

All uni-grams lists $\{UniGrams(p_1), \ldots, UniGrams(p_n)\}$ are then merged to build the uni-gram profile and, similarly, the lists of bi-grams and tri-grams are merged to build the bi-gram and the tri-gram profiles. More specifically, given the lists of uni-grams extracted from the relevant papers, each distinct uni-gram is stemmed and then inserted in the final list of relevant keyphrases. The weight assigned to each uni-gram in the uni-gram profile is computed by summing the weights associated to it by the keyphrase extraction technique. The weight of each keyphrase is then multiplied by the idf value associated to the specific keyphrase. The same technique is applied to produce the bi-gram profile and

the tri-gram profile. Figure 2 shows the most relevant uni-grams, bi-grams and tri-grams extracted from the user profile of one of the users (using a set of 10 relevant documents and removing the uni-grams with a keyphraseness lower the 0.9, the bi-grams with a keyphraseness lower than 0.8 and the tri-grams with a keyphraseness lower than 0.8) in the dataset described in the next section.

Uni-gram profile		Bi-gram profile		Tri-gram profile	
Uni-gram	Weight	Bi-gram	Weight	Tri-gram	Weight
polar	0,35567	movi review	0,65032	span of text	0,69142
sentim	0,24665	discours relat	0,50898	user expertis model	0,57789
movi	0,2432	discours marker	0,45321	inform extract system	0,55485
causal	0,23811	discours process	0,40369	discours relat classifi	0,54869
opinion	0,19106	cue phrase	0,38649	agreem and disagr	0,51035
orient	0,18011	system utter	0,37645	type of discours	0,50749
recommend	0,16508	dialogu act	0,36563	dialogu act type	0,50715
mckeown	0,16225	dialogu system	0,3578	logarithm opinion pool	0,50439
respond	0,1608	extract pattern	0,33828	speech understand result	0,49828
altavista	0,15932	review classif	0,33777	polar discours relat	0,49401

Fig. 2. An example of a user profile. The most relevant n-grams.

In order to compute the relevance of a new paper p_k for a given user profile, the approach follows a similar path: it extracts the three lists of keyphrases from the paper and, then, these keyphrases are stemmed.

The final step is performed by the Matching Module (see Figure 1), which takes in input three lists, $UniGrams(p_k)$, $BiGrams(p_k)$ and $TriGrams(p_k)$, and the user profile. The matching process is based on the cosine similarity producing three similarity values, one for each category of n-grams. Then, an appropriate combination (linear in the first experiments) of these three similarity values is used to compute a unique score to be assigned to the considered paper p_k. Finally, they highest score papers are recommended to the active user.

4 Evaluation

The proposed approach extracts keyphrases from scientific papers in order to have a rich description of user interests which, in turn, is exploited to improve the quality of a content-based recommender system. The main assumption is that keyphrases have meaningful contextual information (not accounted in the classical bag-of-word model) which can be used to improve a cognitive filtering mechanism. In order to validate our claim we performed some experimental evaluation by using a publicly available dataset which contains 597 full papers extracted from the ACL Anthology Reference Corpus (ACL ARC)[6]: this dataset has been built from a significant subset of the ACL Anthology, a scientific digital library of

[6] `http://acl-arc.comp.nus.edu.sg/`

papers on natural language processing and computational linguistics, composed by 10921 papers published since February 2007. The dataset includes specific data about 28 researchers (15 junior researchers and 13 senior researchers), interested in natural language processing. In particular, each researcher reported his relevant papers.

In our evaluation we used this feedback to both build the user profiles of researchers and evaluate the precision of the computed recommendations. More specifically, given a researcher we divided the set of his preferences into two set of papers. We used one of these sets (constituted by 20 papers) as training set in order to build the user profiles while the second set has been used as test set for comparing the results provided by the recommendation engine. According to this setting we computed the recommendations for the researchers in the dataset. Moreover, in order to evaluate the improvement with respect to a baseline bag-of-word approach, we also built the user profiles and document representations using only uni-grams. By using this setting (which does not include bi-grams and tri-grams) we computed a new set of recommendations to be used as a baseline reference for performance.

The following table compares the precision of the recommendations obtained by using only unigrams to the precision obtained by using the approach described in 3.2 where the precision is computed as the ratio among the number of correct recommendations and the number of produced recommendations.

Table 1. The precision of the proposed approach vs. the precision obtained by using only uni-grams

	Uni-grams Based	The Proposed Approach
p@1	0.83	0.93
p@3	0.61	0.77
p@5	0.65	0.80
p@7	0.64	0.73

The table shows that by tanking into account also bi-grams and tri-grams it is possible to obtain an higher precision.

5 Conclusion and Future Work

In this work we introduced a content-based paper recommender system which produces rich user profiles and resource descriptions by extracting keyphrases from scientific articles. The system is intended to work in combination with other modules of an hosting framework, currently under development within Pirates, a larger project aimed at innovate within a social/semantic approach the tools for access, classification, filtering, retrieval, and extraction of Web information. The proposed approach is based on adaptive user profiles and semantic descriptions of resources (documents), which are compared by the cosine similarity to evaluate the relevance of a new document with respect to the user

interests. At the moment, a more extensive evaluation aimed at comparing the proposed approach with other techniques is ongoing. Other future activities will focus on integrating this basic approach with the ontology mining module of the Pirates framework in order to improve the description of both user interests and resources by a deeper semantic representation. Finally, we are currently designing both content-based and collaborative recommender systems able to merge collaborative knowledge (provided from Web 2.0 users) to the semantic knowledge extracted by the Pirates framework.

References

1. Agosti, M.: Information Access Through Digital Library Systems. In: Goh, D.H.-L., Cao, T.H., Sølvberg, I.T., Rasmussen, E. (eds.) ICADL 2007. LNCS, vol. 4822, pp. 11–12. Springer, Heidelberg (2007)
2. Frank, E., Paynter, G.W., Witten, I.H., Gutwin, C., Nevill-Manning, C.G.: Domain-specific keyphrase extraction. In: Proceedings of the 16th International Joint Conference on Artificial Intelligence, Stockholm, Sweden, July 31-August 6, pp. 668–673. Morgan Kaufmann Publishers, San Francisco (1999)
3. Krulwich, B., Burkey, C.: Learning user information interests through the extraction of semantically significant phrases. In: Hearst, M., Hirsh, H. (eds.) AAAI 1996 Spring Symposium on Machine Learning in Information Access, pp. 110–112. AAAI Press, California (1996)
4. Pudota, N., Dattolo, A., Baruzzo, A., Ferrara, F., Tasso, C.: Automatic keyphrase extraction and ontology mining for content-based tag recommendation. International Journal of Intelligent Systems, Special Issue: New Trends for Ontology-Based Knowledge Discovery 25, 1158–1186 (2010)
5. Witten, I.H., Paynter, G.W., Frank, E., Gutwin, C., Nevill-Manning, C.G.: Kea: practical automatic keyphrase extraction. In: Proceedings of the Fourth ACM Conference on Digital Libraries, pp. 254–255. ACM, New York (1999)
6. Hulth, A.: Improved automatic keyword extraction given more linguistic knowledge. In: Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing, pp. 216–223. Association for Computational Linguistics, Morristown (2003)
7. D'Avanzo, E., Magnini, B., Vallin, A.: Keyphrase extraction for summarization purposes: the lake system at duc2004. In: DUC Workshop, Human Language Technology Conference / North American Chapter of the Association for Computational Linguistics Annual Meeting, Boston, USA (2004)
8. Barker, K., Cornacchia, N.: Using Noun Phrase Heads to Extract Document Keyphrases. In: Hamilton, H.J. (ed.) Canadian AI 2000. LNCS (LNAI), vol. 1822, pp. 40–52. Springer, Heidelberg (2000)
9. Bracewell, D.B., Ren, F., Kuroiwa, S.: Multilingual single document keyword extraction for information retrieval. In: Proceedings of the 2005 IEEE International Conference on Natural Language Processing and Knowledge Engineering, Wuhan, pp. 517–522 (2005)
10. Mihalcea, R., Tarau, P.: Textrank: Bringing order into texts. In: Dekang, L., Dekai, W. (eds.) Proc. of Empirical Methods in Natural Language Processing, pp. 404–411. Association for Computational Linguistics, Barcelona (2004)

11. Litvak, M., Last, M.: Graph-based keyword extraction for single-document summarization. In: Proceedings of the Workshop on Multi-source Multilingual Information Extraction and Summarization, pp. 17–24. ACL, Morristown (2008)
12. Brin, S., Page, L.: The anatomy of a large-scale hypertextual web search engine. Computer Networks 30, 107–117 (1998)
13. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE Transaction on Knowledge and Data Engineering 17, 734–749 (2005)
14. Malone, T.W., Grant, K.R., Turbak, F.A., Brobst, S.A., Cohen, M.D.: Intelligent information-sharing systems. Communications of ACM 30, 390–402 (1987)
15. Burke, R.: Hybrid Web Recommender Systems. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) Adaptive Web 2007. LNCS, vol. 4321, pp. 377–408. Springer, Heidelberg (2007)
16. Dattolo, A., Ferrara, F., Tasso, C.: Supporting Personalized User Concept Spaces and Recommendations for a Publication Sharing System. In: Houben, G.-J., McCalla, G., Pianesi, F., Zancanaro, M. (eds.) UMAP 2009. LNCS, vol. 5535, pp. 325–330. Springer, Heidelberg (2009)
17. Chandrasekaran, K., Gauch, S., Lakkaraju, P., Luong, H.P.: Concept-Based Document Recommendations for CiteSeer Authors. In: Nejdl, W., Kay, J., Pu, P., Herder, E. (eds.) AH 2008. LNCS, vol. 5149, pp. 83–92. Springer, Heidelberg (2008)
18. Sugiyama, K., Kan, M.-Y.: Scholarly paper recommendation via user's recent research interests. In: Proceedings of the 10th Annual Joint Conference on Digital Libraries, JCDL 2010, pp. 29–38. ACM, New York (2010)
19. Gori, M., Pucci, A.: Research paper recommender systems: A random-walk based approach. In: Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence, WI 2006, pp. 778–781. IEEE Computer Society, Washington, DC (2006)
20. Baruzzo, A., Dattolo, A., Pudota, N., Tasso, C.: A General Framework for Personalized Text Classification and Annotation. In: Houben, G.-J., McCalla, G., Pianesi, F., Zancanaro, M. (eds.) UMAP 2009. LNCS, vol. 5535, pp. 31–39. Springer, Heidelberg (2009)
21. Dattolo, A., Ferrara, F., Tasso, C.: The Role of Tags for Recommendation: a Survey. In: Hippe, Z., Kulikowski, J., Mroczek, T. (eds.) Backgrounds and Applications 2. AISC. Springer, Heidelberg (in printing)
22. Tasso, C., Asnicar, F.A.: ifweb: a prototype of user model-based intelligent agent for document filtering and navigation in the world wide web. In: 6th UM Inter. Conf. Adaptive Systems and User Modeling on the WWW (1997)
23. Porter, M.F.: An algorithm for suffix stripping. Readings in Information Retrieval, 313–316 (1997)
24. Justeson, J., Katz, S.: Technical terminology: some linguistic properties and an algorithm for identification in text. Natural Language Engineering 1, 9–27 (1995)