A Two-Level Learning Hierarchy of Concept Based Keyword Extraction for Tag Recommendations

Hendri Murfi and Klaus Obermayer

Neural Information Processing Group, TU Berlin
Franklinstr. 28/29, 10587 Berlin, Germany
{henri,oby}@cs.tu-berlin.de
http://ni.cs.tu-berlin.de

Abstract. Textual contents associated to resources are considered as sources of candidate tags to improve the performance of tag recommenders in social tagging systems. In this paper, we propose a two-level learning hierarchy of a concept based keyword extraction method to filter the candidate tags and rank them based on their occurrences in concepts existing in the given resources. Incorporating user-created tags to extract the hidden concept-document relationships distinguishes the two-level from the one-level learning version, which extracts concepts directly using terms existing in textual contents. Our experiment shows that a multi-concept approach, which considers more than one concept for each resource, improves the performance of a single-concept approach, which takes into account just the most relevant concept. Moreover, the experiments also prove that the proposed two-level learning hierarchy gives better performances than one of the one-level version.

Key words: Recommender system, Social tagging, Machine learning, Keyword extraction, Concept extraction

1 Introduction

Social tagging is intended to make resources increasingly easy to discover and recover over time. Discovery enables users to find new content of their interest shared by other users. This social indexing gives a promising index quality because it is done by human beings, who understand the content of the resource, as opposed to software, which algorithmically attempts to determine its meaning. Moreover, it is done collectively among users, that is, it uses a collective human intelligence as an index extractor. Recovery enables a user to recall content that was discovered before. It should be easier because the tags are both originated by, and familiar to, its primary users. However, Golder et al. [9] identify three major problems with the current social tagging systems: polysemy, synonymy, and level variation. The first two inherit the problems of natural language, while the third one refers to the phenomenon of users tagging content at different levels of abstraction. Other problems are dealing with word forms, nouns in singular, nouns in plural, abbreviations, and misspelled words.

To direct users towards the consistency of the tags, the system usually has a service that assists users in the tagging process, by automatically recommending an appropriate set of tags. The service is a mediated suggestion system, that is, the service does not apply the recommended tags automatically, rather it suggests a set of appropriate tags and allows the user to select tags from the set they find appropriate. Moreover, the tag recommendation can serve many purposes such as consolidating the vocabulary across the users, giving a second opinion what a resource is about and, the important thing, increasing the success of searching because of the consistency [14].

In practice, the standard tag recommenders are services that recommend the most popular tags used for either a particular resource or a whole system. There are other methods proposed from a diversity of approaches to recommend tags from user-created tags (folksonomy) such as information retrieval [23, 28], graph-based approaches [11], collaborative filtering [14], machine learning [10, 15]. Recently, people consider textual contents associated to the resources as sources of candidate tags to improve the performance of tag recommenders. For example, Xu et al. [31] suggest content-based (and context-based) tags based on analysis and classification of the tagged content and context. This not only solves the cold start problem, but also increases the tag quality of those objects that are less popular. Tatu et al. [29] use natural language processing tools to extract important terms (nouns, adjectives and named entities) from the textual contents. They conclude that the understanding of the contents improves the quality of the tag recommendations.

In this paper, we also consider the textual contents associated to resources as sources of candidate tags to improve the performance of the tag recommender in the social tagging system. To achieve this goal, we propose a two-level learning hierarchy of concept based keyword extraction as a tag recommendation method. Firstly, the method extracts concepts, which can be considered as a set of related words, using nonnegative matrix factorization (NMF) from training document collections using a two-level learning hierarchy: at the lower level the method extracts concepts and concept-document relationships using user-created tags. Having these relationships, the method populates the concepts with terms existing in textual contents of resources at the higher level. Next, the tag recommender finds the relevant concepts to a given resource and then scales terms of the resource based on their occurrences in the concepts. The terms having the highest scores are set as keywords and recommended as tags. Incorporating the user-created tags to extract the hidden concept-document relationships distinguishes the two-level from the one-level learning version, which extracts concepts directly from terms existing in textual contents. The main advantage of this approach is that NMF algorithm decomposes more compact document representations. Also, the concept extraction from textual contents is handled by nonnegative least squares algorithm which is much more efficient than NMF algorithm. Therefore, the two-level learning hierarchy approach is not only more efficient but also more reliable because it uses tags created by users who understand the content of documents. Moreover, the approach may have richer vocabularies because it can combine vocabularies from tag space and content space. Our experiment shows that a multi-concept approach, which considers more than one concept for each resource, improves the f-measure values of a single-concept approach, which takes into account just the most relevant concept, about 10%. Moreover, the experiments also prove that the proposed two-level learning hierarchy has f-measure values 13% better than one of the one-level version.

The rest of the paper is organized as follows: Section 2 discusses a concept extraction method using nonnegative matrix factorization and our proposed two-level learning hierarcy method. Section 3 describes the existing keyword extraction methods and the proposed concept based keyword extraction methods. In Section 4, we describe a tag recommendation algorithm which combines keywords, which are extracted by the keyword extraction methods, with user-created tags in training data. In Section 5, we show our experiments and results. We conclude and give a summary in Section 6.

2 Concept Extraction

Many researchers are trying to address questions about concepts and, in this section, we consider one of them that defines the concepts as a set of related terms. These definitions are proposed and used by some researchers such as [21] or [27]. They use clustering methods to extract the concepts from training document collections. *Formal concept analysis* (FCA) [5,26] and *Latent semantic analysis* (LSA) [3, 7] are other methods to perform this task .

2.1 A One-Level Learning Hierarchy for Concept Extraction

There are some disadvantages of *singular value decomposition* (SVD) to extract concepts from a document collection as used by LSA. Its negative values make a semantic interpretation difficult. What we would really like to say is that a concept is mostly concerned with some subset of terms, but any semantic interpretation is difficult because of these negative values. To circumvent this problem, a new method which maintains the nonnegative structure of original documents has been proposed. The method uses *nonnegative matrix factorization* (NMF) [17] rather than SVD to extract the concepts from document collections.

Let V be a $m \times n$ term-by-document matrix whose columns are document vectors and a positive integer $k < min(m, n)$. In this paper, we use NMF to extract concepts from the term-by-document matrix V. NMF problem is how to find a nonnegative $m \times k$ matrix W and a nonnegative $k \times n$ matrix H to minimize the functional [2]:

$$\begin{aligned} \min_{W,H} \quad & f\left(W,H\right)=\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n}\left(V_{ij}-\left(WH\right)_{ij}\right)^{2} \\ \text{subject to} \quad & W_{ia}\geq 0, H_{aj}\geq 0, \forall i,a,j \ . \end{aligned} \tag{1}$$

The constrained optimization problem above is convex on either W or H, but not on both, hence realistic possible solutions usually correspond to local minima. The product WH is called a nonnegative matrix factorization of V, although V is not necessarily equal to the product WH. Clearly the product WH is an rank-k approximation to V. An appropriate decision on the value of k is critical in practice, but the choice of k is very often problem dependent. In most cases, however, k is usually chosen such that $k << min(m,n)$.

The most popular approach for the NMF problem is the multiplicative update algorithm proposed by Lee and Seung [18]. To either overcome shortcomings related to convergence properties or to speed up this algorithm, researchers have proposed modifications of the algorithm or even created new ones [2]. In general, the algorithms can be divided into three general classes: multiplicative update algorithms [18,19], gradient descent algorithm [6,12], and alternating least squares algorithms [20, 24].

Because all elements of the matrix W and H are nonnegative, we can interpret them immediately as following: Each column of W corresponds to a set of related terms called concepts and each element w_{ia} of matrix W represents the degree to which term i belongs to concept a. Each element h_{aj} of matrix H represents the degree to which document j is associated to concept a. Next, we call this type of concept extraction as *an one-level learning hierarcy method.*

2.2 A Two-Level Learning Hierarchy for Concept Extraction

In case where the training documents are accompanied by user-created keywords, it is a good idea to incorporate the valuable information in learning process. For this reason, we propose a new learning scheme that uses the keywords for extracting concepts from a document collection. The learning scheme consists of two-level learning hierarchy. At the lower level, concepts and concept-document relationships are discovered using the user-created keywords. Having these relationships, the concepts are populated by terms existing in textual contents of documents at higher level. We expect this method to be successful because the hidden document structures are discovered using keywords collectively created by users. Another advantage of this approach is that NMF algorithm uses more compact document representations. On the other hand, the concept extraction from textual contents is handled by nonnegative least squares algorithm which is much more efficient than NMF algorithm. Therefore, this two-level learning hierarchy approach is not only more efficient but also more reliable because it uses tags created by users who understand the content of documents. Moreover, the approach may have richer vocabularies because it can combine vocabularies from tag space and content space. The detail algorithm of this method is described in Algorithm 1.

Algorithm 1 A two level learning hierarchy for concept extraction

```
1: Let $V$ be the tag by document matrix, and $X$ be the term by document matrix
2: Find the tag by concept matrix $W$ and the concept by document matrix $H$ from
   $V = WH$ using nonnegative matrix factorization (see Section 2.1) to minimize the
   functional:
       $f(W, H) = \frac{1}{2} \|V - WH\|^2$
3: Find the term by concept matrix $T$ from $X = TH$ using nonnegative least squares
   algorithm, e.g. [2]:
     – Solve for $T$ in matrix equation $HH^T T^T = HX^T$
     – Set all negative elements in $T$ to 0
```

3 Concept Based Keyword Extraction

Keyword extraction is the task of automatically selecting a small set of important, topical terms within the textual content of a document. The fact that the keywords are extracted means that the selected terms are present in the document [16]. In general, the task of automatically extracting keywords can be divided into two stages:

1. Selecting candidate terms in the document
2. Filtering out the most significant ones to serve as keywords and rejecting those that are inappropriate

There are various methods proposed for selecting candidate terms. The first one is n-gram extraction, that is, extracting uni-, bi-, or tri-grams, removing those that begin or end with a stop word [8]. Another one is more linguistically oriented using natural language processing (NLP) method such as NP-chunker or part-of-speech (PoS) [13]. Filtering uses either simple statistics, where a weighting schema is applied to rank words accoding to their score [1, 25], or machine learning, where the ranking function is defined by a statistical model derived from training set with manually assigned keywords [13, 22, 30].

In this section, we propose a machine learning based filtering method, that is, a method that uses concepts extracted from textual contents of documents. The method finds the relevant concepts to a given document and then scales terms of the document based on their occurrences in the concepts. The terms having the highest scores are set as keywords. The method can be considered as unsupervised learning when we use the one-level learning hierarchy. It means that the method does not need labeled data for the training process. Moreover, the method becomes a supervised method when the user-created tags are used in learning process for the two-level learning hierarchy approach. Two variants of the concept based keyword extraction method are described in detail in the following sections.

3.1 Single-Concept Based Keyword Extraction

The two-level learning hierarchy extracts concepts from a training document collection. Having these concepts, the single-concept based keyword extraction method finds the most relevant concept to a given document and then scales the candidate terms existing in the document based on their occurrence in the concept. The relevance of a concept c with a document d is calculated using the following cosine distance measure:

$$\mathrm{rel}(\mathbf{d}, c) = \frac{\mathbf{d}^T V_c}{\|\mathbf{d}\| \, \|V_c\|} \tag{2}$$

The detail algorithm of this approach is described in Algorithm 2.

Algorithm 2 The single-concept based keyword extraction

```
Let column $c$ of $W$ $(W_c)$ be concept $c$ and $\mathbf{d}$ be a document
Let $rel(\mathbf{d}, c) = \frac{\mathbf{d}^T W_c}{\|\mathbf{d}\| \|W_c\|}$
Find the most relevant concept to the document $\mathbf{d}$, i.e., concept $c'$ where $c' = \mathrm{argmax}_c(rel(\mathbf{d}, c))$
Scale terms existed in the document based on the most relevant concept, i.e.,
    $d1_i = w_{ic'} d_i$
    $d2_j = t_{jc'} d_j$
Combine the normalized terms:
    $\tilde{\mathbf{d}} = (1 - \alpha) (\mathbf{d1} / \|\mathbf{d1}\|) + \alpha(\mathbf{d2} / \|\mathbf{d2}\|)$
Select first $n$ non zero terms of the ranked $\tilde{\mathbf{d}}$ as keywords
```

3.2 Multi-Concept Based Keyword Extraction

The multi-concept based keyword extraction assumes that a document may contain more than one relevant concept. The detail algorithm of the multi-concept based keyword extraction method is described in Algorithm 3.

4 A Hybrid Tag Recommender

In our experiment, we use a hybrid recommender as described in detail in Algorithm 4. The recommender checks if a given resource exists in the training data.

Algorithm 3 The multi-concept based keyword extraction

```
Let column $c$ of $W$ $(W_c)$ be concept $c$ and $\mathbf{d}$ be a document
Let $rel(\mathbf{d}, c) = \frac{\mathbf{d}^T W_c}{\|\mathbf{d}\| \|W_c\|}$
Scale terms existed in the document using the concepts, i.e.,
    $d1_i = \sum_c \mathrm{rel}(\mathbf{d}, c) w_{ic} d_i$
    $d2_j = \sum_c \mathrm{rel}(\mathbf{d}, c) t_{jc} d_j$
Combine the normalized terms:
    $\tilde{\mathbf{d}} = (1 - \alpha) (\mathbf{d1} / \|\mathbf{d1}\|) + \alpha(\mathbf{d2} / \|\mathbf{d2}\|)$
Select first $n$ non zero terms of the ranked $\tilde{\mathbf{d}}$ as keywords
```

If this is the case then the tag space based recommenders are suggested. The collaborative recommender is used if a given user has profiles in system. Otherwise, the most popular tag by resource method is used as tag recommender. If the resource appears for the first time then the recommender examines the content of the resource using the concept based keyword extraction algorithm. Boosting the extracted tags if they have been used by the user before. If neither any tags nor any keywords are suggested then the most popular tags in the training data are recommended. Using the user-created tags aims to direct the standardization and consistency of supplied tags, while using the tags extracted from textual contents intend especially to overcome the cold start problem.

Algorithm 4 A hybrid tag recommendation algorithm

```
input : a post $P < user, resource >$
if $P.resource$ exists in system then
    if $P.user$ exists in system then
        $P.tags \Leftarrow$ collaborative tags by $P.user$
    else
        $P.tags \Leftarrow$ most popular tags by $P.resource$
    end if
else
    $P.tags \Leftarrow$ the keywords by $P.resource$
    Boosting tags of $P.tags$ that exist in tag space of $P.user$
end if
if $P.tags = \phi$ then
    $P.tags \Leftarrow$ most popular tags
end if
Select top $n$ of the ranked $P.tags$ as recommended tags
```

5 Experiment

We apply our proposed recommender methods (Algorithm 4) for ECML PKDD Discovery Challenge 2009[1]. The task of the competition requires the development of a content-based tag recommendation method for BibSonomy[2], a web based social bookmarking system that enables users to tag both web pages (bookmark) and scientific publications (bibtex). The organizers of the competition made available a training set of examples consisting of the resources accompanied with their user-created tags. A testing data will be provided in order to evaluate proposed recommenders. Each bookmark is described by its URL, a description of the URL that usually is the title of the web page and an extended description of the bookmark supplied by the user. Each bibtex is associated with values of bibtex fields such as title, author, booktitle, journal, series, volume, number, etc. BibtexKey, bibtexAbstract, URL, and description of the publication can be specified. Some statistics of the data are shown in Table 1.

Table 1. Statistics of Experiment data

	Bookmark	Bibtex
Training	181,491	72,124
Testing	16,898	26,104

In our experiment, we use textual contents associated to each resource as content of the resources. For the bookmark, the contents are the description of the URL and the extended description. Title and abstract are textual contents associated to the bibtex. A bookmark is identified by its URL address (*url_hash*) attribute and a bibtex by its title (*simhash1*) attribute. Therefore, a document, bookmark or bibtex, is represented by the description given to the document by all users that bookmarked the document.

Let D be a testing data set, consisting of $|D|$ examples $(\mathbf{r}_i, T_i), i = 1...|D|$. Let T_i be the set of tags created by users for a resource $\mathbf{r}_i$ and P_i be the set of tags predicted by a recommender for a resource $\mathbf{r}_i$. The precision, recall, and F-measure for recommender f on testing data set D is calculated as follows:

$$\text{Precision} = \frac{1}{|D|}\sum_{i=1}^{|D|}\frac{|T_i \cap P_i|}{|P_i|}$$

$$\text{Recall} = \frac{1}{|D|}\sum_{i=1}^{|D|}\frac{|T_i \cap P_i|}{|T_i|}$$

$$\text{F} - \text{Measure} = \frac{1}{|D|}\sum_{i=1}^{|D|}\frac{2\,|T_i \cap P_i|}{|P_i| + |T_i|}$$

[1] http://www.kde.cs.uni-kassel.de/ws/dc09

[2] http://www.bibsonomy.org

We perform our experiment in java platform and use Lucene[3] for creating the tag-by-resource matrix and the term-by-resource matrix. The other processes are conducted on the Weka[4] framework, an open source machine learning software.

5.1 Experiment Settings

For each of the method of our experiment the settings we used to run them are described as following:

Concept-based keyword extraction . For creating the term-by-resource matrix, resources are parsed and a dictionary of terms is created using a standard word tokenization method. The terms are words, special characters are removed, and Snowball Porter stemming and standard stop words of English and German are applied. Finally, the term-by-resource matrix is created using a term frequency weighting scheme.

Extracting concepts from the term-by-resource matrix is an important step to find keywords from new resources. The optimal number of concepts (k), which captures most concepts in the training document collection, remains difficult to find. The method that is usually used for a practical purpose is a heuristic approach. However, because of the memory usage, simulations are usually conducted on the maximum number of concepts that can be extracted. In our experiment, we extract 200 concepts for the training document collection. For this task, we use the nonnegative double SVD initialization method [4] that conducts no randomization and the projected gradient method [20] that converges to a local minimum. We expect these combining methods leads to converge to a unique solution with a minimum error.

There is another parameter α that should be optimized in the two-level learning hierarchy approach. The parameter reflects the portion of the tag space and the content space as sources of tags for the recommender. In our experiment, we set the parameter $\alpha = 0.25$ for the single-concept method and $\alpha = 0.05$ for the multi-concept method, which are the optimal values we get using a heuristic method.

Collaborative recommendation [14] . For a given tag-by-user matrix X, a given user u, a given resource r, and integer k and n, the set $T(u, r)$ of n recommended tags is calculated by:

$$T(u, r) = \text{argmax}^n_{t \in T} \sum_{v \in N^k_u} sim(X_u, X_v) \delta(v, t, r) \tag{3}$$

where N^k_u is k nearest neighbors of u in X, $\delta(v, t, r) = 1$ if $(v, t, r) \in folksonomy$ and 0 else. Therefore, the only parameter to be tuned is the number of neighbors k. For that, multiple runs where performed where k incremented until a point where no more improvement in the results were observed.

[3] http://lucene.apache.org/

[4] http://www.cs.waikato.ac.nz/ml/weka/

Most popular tags by resource . For a given resource we count how many posts a tag occur together with that resource. We use tags that occur most often together with that resource as recommendation.

Most popular tags . For each tags we count in how many posts it occurs. We then use tags that occur most often as recommendation.

5.2 Single- vs. Multi-Concept Method

Fig. 1 shows the performances of the single- and multi-concept based keyword extraction on testing data. From Fig. 1, we can calculate that recall, precision, and f-measure of the multi-concept approach are, on average, 10%, 15% and 12%. The recall is likely to increase when the number of recommended tags gets bigger, while the precision is reduced for the bigger numbers of tags. Fig. 1 also shows the performance of the single-concept approach in the similar pattern and its f-measure is, on average, 11%. From both curves, we conclude that the multi-concept approach, which assumes that a resource may contain more than one concept, improves f-measure of the single-concept method, on average, 10%. The improvement occurs in all numbers of recommended tags. These results verify that associating of resources with more than one concept gives better performance than just considering the main concept of resources. In other words, some minor concepts of a resource should also be examined for getting the better performance of the keyword extraction.

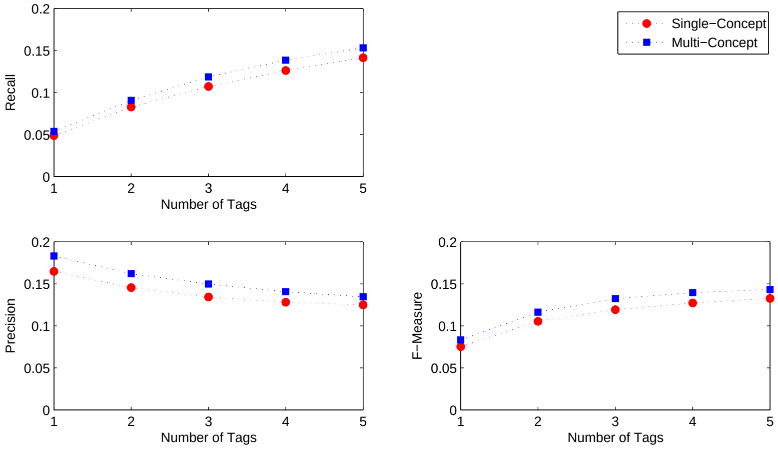

Fig. 1. Performance comparison of single- and multi-concept approach

5.3 One- vs. Two-Level Learning Hierarchy

In this section, we examine the performance of our proposed two-level learning hierarchy approach compared to the one-level version. Fig. 2 shows the performance of the one-level learning hierarchy multi-concept based keyword extraction and the two-level learning hierarchy multi-concept based keyword extraction. From the figure, we see that the two-level learning method has better recall, precision and f-measure. Its f-measure values, on average, are 13% better than one of the one-level learning approach. The detailed recall, precision, and f-measure values of the optimal performance of the two-level learning hierarchy are given in Table 2.

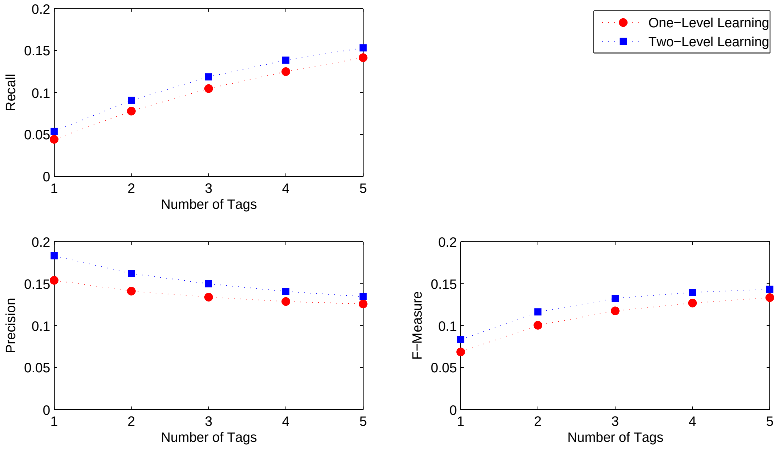

Fig. 2. Performance comparison of one- and two-level learning hierarchy approach

Table 2. Performance of two-level learning hierarchy of multi-concept based keyword extraction method for each number of recommended tags using the optimal parameter $\alpha = 0.05$

Num. of Tags	Recall	Precision	F-Measure
1	0.0538	0.1832	0.0832
2	0.0908	0.1621	0.1164
3	0.1187	0.1498	0.1324
4	0.1386	0.1406	0.1396
5	0.1533	0.1345	0.1433

6 Summary

In this paper, we propose a two-level learning hierarchy concept based keyword extraction method for task1 of ECML PKDD Discovery Challenge 2009, that is, a content-based tag recommendation. The tag recommendation method explores tags from textual contents of resources using concepts existing in the textual contents of the resources. A multi-concept approach, which considers more than one concept for each resource, improves the performance of a single-concept approach, which only considers the most relevant concept. Moreover, our experiment demonstrates that the proposed two-level learning hierarchy method outperforms the common one-level learning approach for all performance measures, e.g. recall, precision, and f-measure.

7 Acknowledgments

The authors would like to thank Nicolas Neubauer for useful discussions, comments and suggestions in writing this paper. The first author was funded partly by a scholarship by the DAAD.

References

1. K. Barker and N. Cornacchia. Using noun phrase heads to extract document keyphrases. In *Proceedings of the 13th Canadian Conference on Artificial Intelligence*, pages 417–426, Montreal, Canada, 2000.
2. M. W. Berry, M. Browne, A. N. Langville, P. Pauca, and R. J. Plemmons. Algorithms and applications for approximate nonnegative matrix factorization. *Computational Statistics and Data Analysis*, 15(1):155–173, 2007.
3. W. M. Berry, S. T. Dumais, and G. W. O'Brien. Using linear algebra for intelligent information retrieval. *SIAM Review*, 37:573–595, 1995.
4. C. Boutsidis and E. Gallopoulos. Svd-based initialization: A head start on nonnegative matrix factorization. *Pattern Recognition*, 41(4):1350–1362, 2008.
5. C. Carpineto and G. Romano. Using concept lattices for text retrieval and mining. In B. G. et al., editor, *Formal Concept Analysis*, pages 161–179. Springer-Verlag, 2005.
6. M. Chu, F. Diele, R. Plemmons, and S. Ragni. Optimality, computation, and interpretations of nonnegative matrix factorization. Technical report, Department of Mathematics, North Carolina State University, 2005.
7. S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407, 1990.
8. E. Frank, G. W. Paynter, I. H. Witten, C. Gutwin, and C. G. Nevill-Manning. Domain-specific keyphrase extraction. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 668–673, 1999.
9. S. Golder and B. Huberman. Usage patterns of collaborative tagging systems. *Journal of Information Science*, 32(2):198–208, 2006.
10. P. Heymann, D. Ramage, and H. G. Molina. Social tag prediction. In *Proceeding of SIGIR*, 2008.

11. A. Hotho, R. Jaschke, C. Schmitz, and G. Stumme. Information retrieval in folksonomies: Search and ranking. In *Proceeding of the 3rd European Semantic Web Conference*, pages 411–426, 2006.
12. P. Hoyer. Non-negative matrix factorization with sparseness constraints. *Journal of Machine Learning Research*, 5:1457–1469, 1994.
13. A. Hulth. *Combining Machine Learning and Natural Language Processing for Automatic Keyword Extraction.* PhD thesis, Stockholm University, 2004.
14. R. Jaschke, L. Marinho, A. Hotho, L. S. Thieme, and G. Stumme. Tag recommendations in folksonomies. In J. N. Kok, J. Koronacki, R. L. de Ma'ntaras, S. Matwin, D. Mladenic, and A. Skowron, editors, *The 11th European Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 506–514, 2007.
15. I. Katakis, G. Tsoumakas, and I. Vlahavas. Multilabel text classification for automated tag suggestion. In *Proceeding of the ECML/PKDD Discovery Challenge*, 2008.
16. F. W. Lancaster. *Indexing and Abstracting in Theory and Practice.* Library Association Publishing, 1998.
17. D. D. Lee and H. S. Seung. Learning the parts of objects by nonnegative matrix factorization. *Nature*, 401:788–791, 1999.
18. D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing System*, pages 556–562, 2001.
19. C. J. Lin. On the convergence of multiplicative update algorithms for non-negative matrix factorization. *IEEE Transactions on Neural Networks*, 18:1589–1596, 2007.
20. C. J. Lin. Projected gradient methods for non-negative matrix factorization. *Neural Computation*, 19:2756–2779, 2007.
21. D. Lin and P. Pantel. Concept discovery from text. In *Proceedings of the International Conference on Computational Linguistics*, pages 577–583, 2002.
22. O. Medelyan and I. H. Witten. Domain-independent automatic keyphrase indexing with small training set. *Journal of the American Society for Information Science and Technology*, 59(7):1026–1040, 2008.
23. G. Mishne. Autotag: A collaborative approach to automated tag assignment for weblog posts. In *Proceeding of the 15th International Conference on World Wide Web*, pages 953–954, 2006.
24. P. Paatero and U. Tapper. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5:111–126, 1994.
25. C. Paice and W. Black. A three-pronged approach to the extraction of key terms and semantic roles. In *Proceedings of the International Conference on Recent Advances in NLP*, pages 357–363, Borovets, Bulgaria, 2003.
26. U. Priss. Formal concept analysis in information science. *Annual Review of Information Science and Technology*, 40:521–543, 2005.
27. M. L. Reinberger and P. Spyns. Unsupervised text learning for the learning of dogma-inspired ontologies. In P. Buitelaar, P. Cimiano, and B. Magnini, editors, *Ontology Learning from Text: Method, Application and Evaluation*, pages 29–43. IOS Press, 2005.
28. S. C. Sood, S. H. Owsley, K. J. Hammond, and L. Birnbaum. Tagassist: Automatic tag suggestion for blog posts. In *Proceeding of the International Conference on Weblogs and Social Media (ICWSM)*, 2007.
29. M. Tatu, M. Srikanth, and T. D'Silva. Rsdc'08: Tag recommendation using bookmark content. In *Proceeding of the ECML/PKDD Discovery Challenge*, 2008.

30. P. D. Turney. Learning algorithms for keyphrase extraction. *Information Retrieval*, 2(4):303–336, 2000.
31. Z. Xu, Y. Fu, J. Mao, and D. Su. Towards the semantic web: Collaborative tag suggestions. In *Proceeding of Collaborative Web Tagging Workshop*, 2006.