SEMI-AUTOMATIC CONSTRUCTION OF TOPIC ONTOLOGY

Blaž Fortuna, Dunja Mladenić, Marko Grobelnik
Department of Knowledge Technologies
Jozef Stefan Institute
Jamova 39, 1000 Ljubljana, Slovenia
Tel: +386 1 4773419; fax: +386 1 4251038
e-mail: blaz.fortuna@ijs.si

ABSTRACT

In this paper, we review two techniques for topic discovery in collections of text documents (Latent Semantic Indexing and K-Means clustering) and present how we integrated them into a system for semi-automatic topic ontology construction. The system offers supports to the user during the construction process by suggesting topics and analysing them in real time.

1 INTRODUCTION

When working with large corpora of documents it is difficult to comprehend and process all the information contained in them. Standard text mining and information retrieval techniques usually rely on word matching and do not take into account the similarity of words and structure of documents within the corpus. We try to overcome that by automatically extracting the topics covered within the documents in the coprus and helping the user to organize them into a topic ontology.

Topic ontology is a set of topics connected with different types of relations. Each topic includes a set of related documents. Construction of such an ontology from a given corpus can be a very time consuming task for the user. In order to get a feeling on what the topics in the corpus are, what the relations between topics are and, at the end, to assign each document to some certain topics, the user has to go through all the documents. We tried to overcame this by building a special tool which helps the user by suggesting the possible new topics and visualizing the topic ontology created so far –- all in real time. This tool in combination with the corpus visualization tools [3] aims at assisting the user in a fast semi-automatic construction of the topic ontology from a large document collection.

We chose two different approaches for discovering topics within the corpora. The first approach is a linear dimensionality reduction technique, known as Latent Semantic Indexing (LSI) [2]. This technique relies on the fact that words related to the same topic co-occur together more often than words related to the different topics. The result of LSI are fuzzy clusters of words each describing one topic. The second approach we used for extracting topics is a well known k-means clustering algorithm [6]. It partitions the corpus into *k* clusters so that two documents within the same cluster are more closely related than two documents from different clusters. We used this two algorithms for automatic suggestion of topics during the construction of the topic ontology.

This paper is organized as follows. Section 2 gives a short overview of the related work on building ontologies. Section 3 gives an introduction to the text mining techniques we used. Details about our system are presented in Section 4, followed by the conclusions in Section 5.

2 TEXT MINING TECHNIQUES

2.1 Representation of text documents

In order to use the algorithms we will describe later we must first represent text documents as vectors. We use standard *Bag-of-Words* (BOW) approach together with the TFIDF weighting [8]. This representation is often referred to as a *vector-space model*. The similarity between two documents is defined as the cosine of the angle between their vector representations – *cosine similarity*. Note that the cosine similarity between two exactly the same documents is 1 and the similarity between two documents that share no common words is 0.

2.2 Latent Semantic Indexing

Latent Semantic Indexing (LSI) is a technique for extracting this background knowledge from text documents. It uses a technique from linear algebra called Singular Value Decomposition (SVD) and bag-of-words representation of text documents for extracting words with similar meanings. This can also be viewed as extraction of hidden semantic concepts or topics from text documents.

Most well known and used approach is Latent Semantic Indexing as described in [2]. First term-document matrix A is constructed from a given set of text documents. This is a matrix with bag-of-words vectors of documents as columns.

This matrix is decomposed using singular value decomposition so that A = USVT where matrices U and V are orthogonal and S is a diagonal matrix with ordered singular values on the diagonal. Columns of matrix U form an orthogonal basis of a subspace in bag-of-words space where vectors with higher singular values carry more information (this follows from theorem that by truncating singular values to only biggest *k* we get the best approximation for matrix A with rank *k*). Because of all this, vectors that form the basis can also be viewed as concepts or topics. The space spanned by these vectors is called Semantic Space.

2.3 K-Means clustering

Clustering is a technique for partitioning data so that each partition (or cluster) contains only points which are similar according to some predefined metric. In the case of text this can be seen as finding groups of similar documents, that is documents which share similar words.
K-Means [6] is an iterative algorithm which partitions the data into *k* clusters. It has already been successfully used on text documents [9] to cluster a large document corpus based on the document topic and incorporated in an approach for visualizing a large document collection [4].

2.4 Keyword extraction

We used two methods for extracting keywords from a given set of documents: (1) keyword extraction using centroid vectos and (2) keyword extraction using SVM [1]. We used this two methods to generate description for a given topic based on the documents inside the topic.

The first method works by using the centroid vector of the topic (centroid is the sum of all the vectors of the document inside the topic). The main keywords are selected to be the words with the highest weights in the centroid vector.

The second method is based on the idea presented in [1] which uses Support Vector Machine (SVM) binary classifier [7]. Let *A* be the topic which we want to describe with keywords. We take all the documents from the topics that have *A* for a subtopic and mark these documents as negative. We take all the documents from the topic *A* and mark them as positive. If one document is assigned both negative and positive label we say it is positive. Then we learn a linear SVM classifiers on these documents and classify the centroid of the topic *A*. Keywords describing *A* are the words, which's weights in SVM normal contribute most when deciding if centroid is positive.

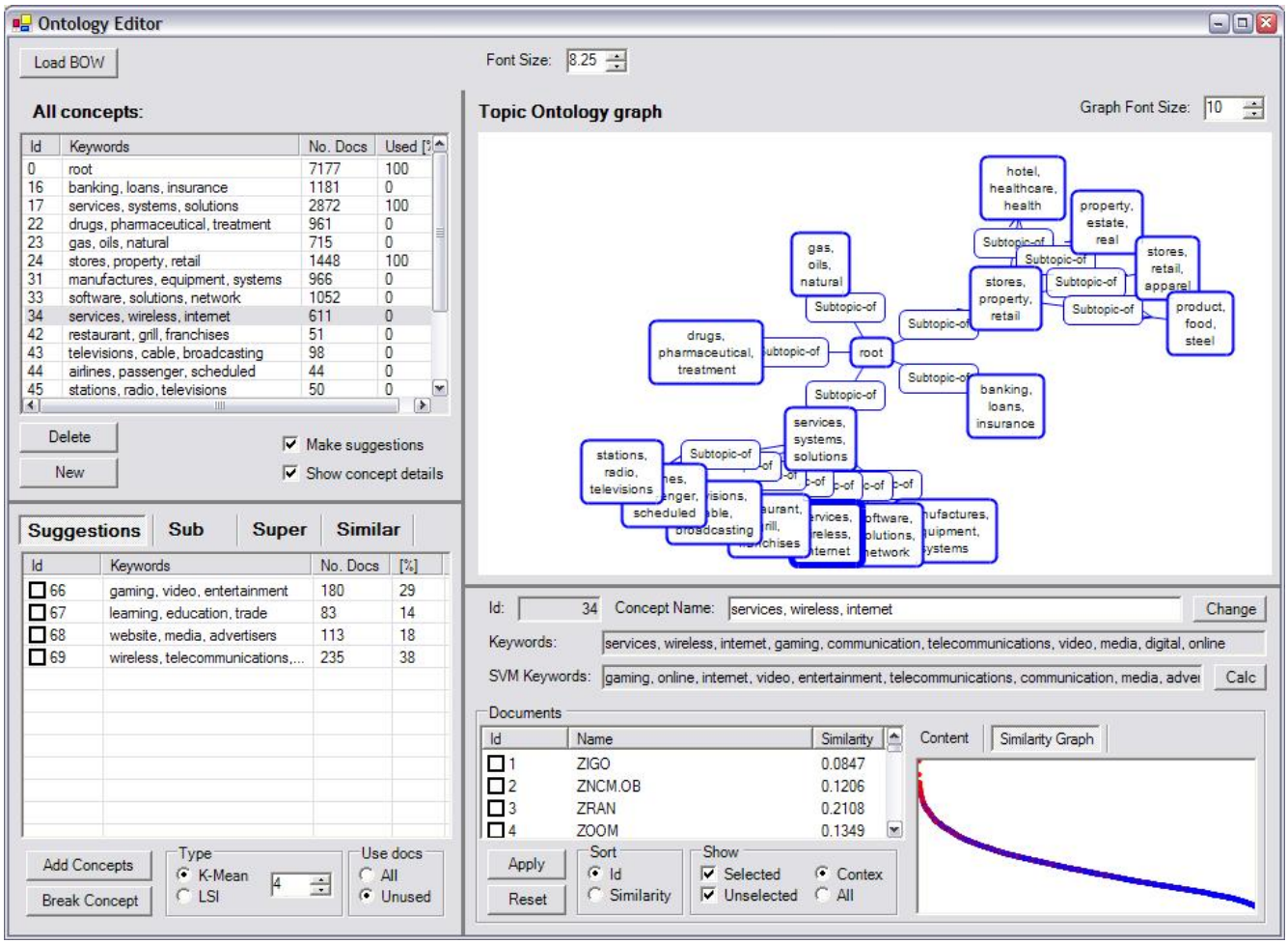

Figure 1 *Screen shot of the interactive system for construction topic ontologies.*

The difference between these two approaches is that the second approach takes into account the context of the topic. Let's say that we have a topic named `computers'. When deciding, what the keywords for some subtopic *A* are, the first method would only look at what the most important words within the subtopic *A* are and words like `computer' would most probably be found important. However, we already know that *A* is a subtopic of `computers' and we are more interested in finding the keywords that separate it from the other documents within the `computers' topic. The second method does that by taking the documents from all the super-topics of *A* as a context and learns the most crucial words using SVM..

3 SEMI-AUTOMATIC CONSTRUCTION OF TOPIC ONTOLOGY

We view semi-automatic topic ontology construction as a process where the user is taking all the decisions while the computer only gives suggestions for the topics, helps by automatically assigning documents to the topics, helps by suggestiong names for the topics, etc. The suggestions are applied only when the users decides to do so. The computer also helps by visualizing the topic ontology and the documents.

In Figure 1 you can see the main window of the interactive system we developed. The system has three major parts that will be further discussed in following subsections. In the central part of the main window is a visalization of the current topic ontology (Ontology visualization). On the left side of the window is a list of all the topics from this ontology. Here the user can select the topic he wants to edit or further expand into subtopics. Further down is the list of suggested subtopics for the selected topic (Topic suggestion) and the list with all topics that are in relation-ship with the selected topic. At the bottom side of the window is the place where the user can fine-tune the selected topic (Topic management).

Figure 2 *Example of topic ontology visualization.*

3.1 Ontology visualization

While the user is constructing/changing topic ontology, the system visualizes it in real time as a graph with topics as nodes and relations between topics as edges. See Figure 2 for an example of the visualization.

3.2 Topic suggestion

When the user selects a topic, the system automatically suggests possible subtopics of the selected topic. This is done by LSI or k-means algorithms applied only to the documents from the selected topic. The number of suggested topics is supervised by the user. Then, the user selects the subtopics s/he finds reasonable and the system automatically adds them to the ontology with relation `*subtopic-of*' to the selected topic. The user can also decide to replace the selected topic with the suggested subtopics. Figure 3 shows this feature implemented in our system.

3.3 Topic management

The user can manually edit each of the topics s/he added to the topic ontology. The user can change which documents

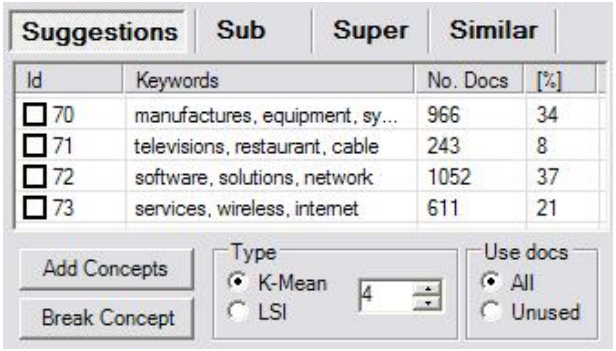

Figure 3 *Example of suggested subtopics.*

are assigned to this topic (one document can belong to more topics), the name of the topic and relationship of the topic to other topics. The main relationship is *subtopic-of* and is automatically added when adding subtopics as described in the previous section. The user can control all the relations between topics by adding, removing, directing and naming the relations.

Here the system can provide help on more levels (see Fgiure 4 for details):

- The system automatically assigns the documents to a topic when it is added to the ontology.
- The system helps by providing the keywords describing the topic using the methods described in Section 3. This can assist user when naming the topic.
- The system computes the cosine similarity between each document from the corpus and the centroid of the topic. This information can assist the user when searching for documents related to the topic. The similarity is shown on the list of documents next to the document name and the graph of similarities is ploted next to the list.

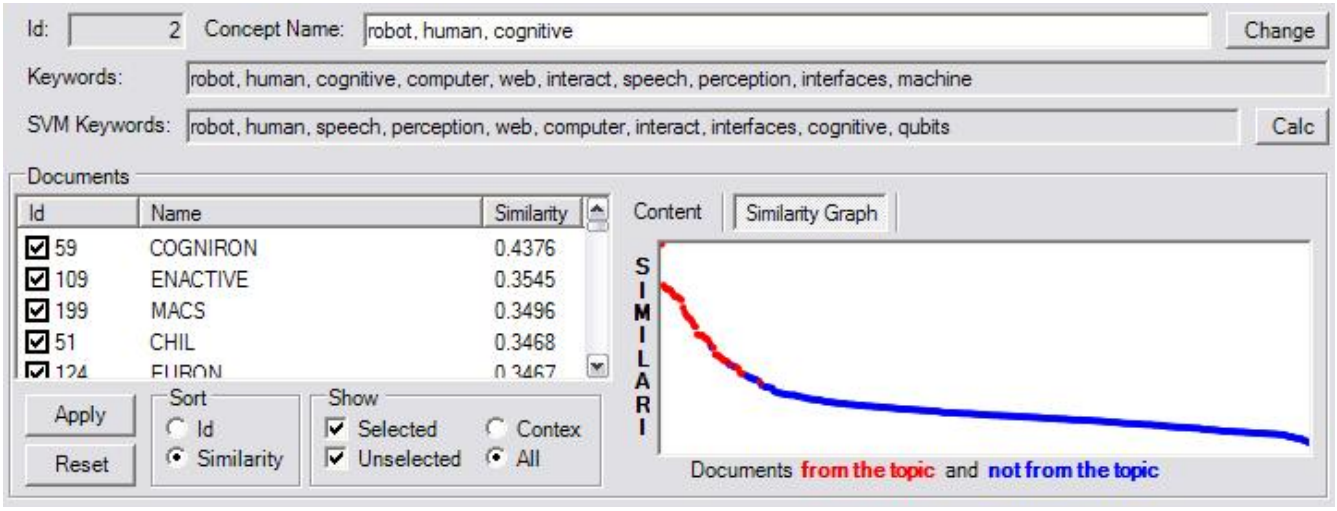

Figure 4 *Topic management.*

- The system also computes similarities between the selected topic and all the other topics from the ontology. For the similarity measure between two topics it uses either the cosine similarity between their centroid vectors or the intersection between their documents.

4 CONCLUSION

In this paper we presented our approach to the semi-automatic construction of topic ontologies. In the first part of the paper we presented text mining techniques we used: two methods for discovering topics within the corpus, LSI and K-Means clustering, and two methods for extracting keywors. In the second part we showed how we integrated all these methods into an interactive system for constructing topic ontologies.

Since this is work-in-progress there is a large area of possible improvements. The most important next step is to evaluate the proposed system in some practical scenarios and see how it fits the needs of the users and what features are missing or need improvement. Another possible direction is making the whole process more automatic and reduce the need for user interaction. This involves things like calculating the quality of topics suggested by the system, more automated discovery of the optimal number of topics, more support for annotating the documents with the topics, discovering different kinds of relations between topics etc.

We would also like to explore other techniques for concept/topic discovery (for example Probabilistic Latent Semantic Analysis [5] and its derivates) and are considering possible integrations with other tools for ontology building and management.

Acknowledgement

This work was supported by the Slovenian Research Agency and the IST Programme of the European Community under SEKT Semantically Enabled Knowledge Technologies (IST-1-506826-IP) and PASCAL Network of Excellence (IST-2002-506778). This publication only reflects the authors' views.

References

[1] Brank, J., Grobelnik, M., Milic-Frayling, N., Mladenic, D.: Feature selection using support vector machines. *Proceedings of the Third International Conference on Data Mining Methods and Databases for Engineering, Finance, and Other Fields, Bologna, Italy, 25--27 September 2002*.

[2] Deerwester, S., Dumais, S., Furnas, G., Landuer, T., And Harshman, R. (1990): *Indexing by Latent Semantic Analysis, , Journal of the American Society of Information Science, vol. 41, no. 6, 391-407*

[3] Fortuna, B., Grobelnik, M., Mladenic, D. (2005): Visualization of text document coprus. *Proceedings of the 8th International multi-conference Information Society IS-2005, Ljubljana: Institut Jozef Stefan.*

[4] Grobelnik, M., And Mladenic, D. (2002): Efficient visualization of large text corpora. *Proceedings of the Seventh TELRI seminar. Dubrovnik, Croatia*

[5] Hoffman, T. (1999): Probabilistic Latent Semantic Analysis, *Proc. of Uncertainty in Artificial Intelligence, UAI'99*

[6] Jain, Murty and Flynn (1999): Data Clustering: A Review, *ACM Comp. Surv.*

[7] T. Joachims (1999): Making large-scale svm learning practical. *In B. Scholkopf, C. Burges, and A. Smola, editors, Advances in Kernel Methods - Support Vector Learning. MIT-Press.*

[8] Salton, G. (1991): Developments in Automatic Text Retrieval, *Science, Vol 253, pages 974-979*

[9] Steinbach, M., Karypis, G., Kumar, V. (2000): A comparison of document clustering techniques. In Proceedings Of KDD Workshop On Text Mining, Pp. 109–110