Semi-automatic construction of topic ontology

Blaž Fortuna¹, Dunja Mladenić¹, and Marko Grobelnik¹

Jožef Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia,
blaz.fortuna@ijs.si, dunja.mladenic@ijs.si, marko.grobelnik@ijs.si
WWW home page: http://kt.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 support 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 hard 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 the structure of the documents within the corpus. We try to overcome that by automatically extracting the topics covered within the documents in the corpus 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 overcome 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 [8] 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) [5]. 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 [12]. It partitions the corpus into k clusters so that two documents within the same cluster are more closely related than two documents from two 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 Related work on building ontologies

Different approaches have been used for building ontologies, most of them using mainly manual methods. An approach to building ontologies was set up in the CYC project [6], where the main step involved manual extraction of common sense knowledge from different sources. There have been some definitions of methodology for building ontologies, again assuming manual approach. For instance, the methodology proposed in [19] involves the following stages: identifying the purpose of the ontology (why to build it, how will it be used, range of the users), building the ontology, evaluation and documentation. The building of the ontology is further divided in three steps. The first is ontology capture, where key concepts and relationships are identified, a precise textual definition of them is written, terms to be used to refer to the concepts and relations are identified, the involved actors agree on the definitions and terms. The second step involves coding of the ontology to represent the defined conceptualization in some formal languages (committing to some meta-ontology, choosing a representation language and coding). The third step involves possible integration with existing ontologies. An overview of the methodologies for building ontologies is provided in [7], where several methodologies, including the above described one, are presented and analyzed against the IEEE Standard for Developing Software Life Cycle Processes viewing ontologies as parts of some software product.

Recently, a number of workshops at Artificial Intelligence and Machine Learning conferences (ECAI, IJCAI, ECML/PKDD) have been organized on learning ontologies. Most of the work presented there addresses one of the following: a problem of extending an existing ontology WordNet using Web documents [1], using clustering for semi-automatic construction of ontologies from parsed text corpora [2], [16], learning taxonomic, eg., isa, [4], and non-taxonomic, eg., has-Part relations [15], extracting semantic relations from text based on collocations [10], extracting semantic graphs from text for learning summaries [14].

The contribution of this paper to the field is that it presents a novel approach to semi-automatic construction of a very specific type of ontology – topic ontology. The system we developed helps the user at the first out of the three steps from the methodology [19]. Text mining techniques (e.g. clustering) were already proven successful when used at this step (e.g. [2], [16]) and in this paper we presents a very tight integration of them with a manual ontology construction tool. This allows our system to offer support to the user during the whole ontology construction process.
3 Text mining techniques

3.1 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 [17]. This representation is often referred to as 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.

3.2 Latent Semantic Indexing

The language contains many redundant information, since many words share common or similar meaning. For computer this can be difficult to handle without some additional information – background knowledge. Latent Semantic Indexing (LSI), [5], 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 detecting words with similar meanings. This can also be viewed as extraction of hidden semantic concepts or topics from the text documents.

3.3 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 [12] is an iterative algorithm which partitions the data into $k$ clusters. It has already been successfully used on text documents [18] to cluster a large document corpus based on the document topic and incorporated in an approach for visualizing a large document collection [9].

3.4 Keywords extraction

We used two methods for extracting keywords from a given set of documents: (1) keyword extraction using centroid vectors and (2) keyword extraction using SVM [3]. 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 [3] which uses Support Vector Machine (SVM) binary classifier [13]. 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 the concept A are the words, which’s weights in SVM normal vector contribute most when deciding if centroid is positive.

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.

4 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 suggestion 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 visualization 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).

4.1 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.

4.2 Topic suggestion

When the user selects a topic, the system automatically suggests what the subtopics of the selected topic could be. This is done by LSI or k-means al-
Fig. 1. Screen shot of the interactive system for construction topic ontologies.

Fig. 2. Example of topic ontology visualization.
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 he finds reasonable and the system automatically adds them to the ontology with relation ‘subtopic-of’ to the selected topic. User can also decide to replace the selected topic with the suggested subtopics. In Figure 3 you can see how is this feature implemented in our system.

<table>
<thead>
<tr>
<th>Suggestions</th>
<th>Sub</th>
<th>Super</th>
<th>Similar</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Keywords</td>
<td>No. Docs</td>
<td>[%]</td>
</tr>
<tr>
<td>70</td>
<td>manufacture, equipment, c...</td>
<td>966</td>
<td>34</td>
</tr>
<tr>
<td>71</td>
<td>televisions, restaurant, cable</td>
<td>243</td>
<td>8</td>
</tr>
<tr>
<td>72</td>
<td>software, solutions, network</td>
<td>1052</td>
<td>37</td>
</tr>
<tr>
<td>73</td>
<td>services, wireless, internet</td>
<td>611</td>
<td>21</td>
</tr>
</tbody>
</table>

Add Concepts | Break Concept

Type
- K-Mean
- LSI

Use docs
- All
- Unused

Fig. 3. Example of suggested subtopics.

4.3 Topic management

The user can manually edit each of the topics he added to the topic ontology. He can change which documents are assigned to this topic (one document can belong to more topics), what is the name of the topic and what is the 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:

- 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 plotted next to the list.
- 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.
See Figure 4 for details on how this is integrated into our system.

5 Conclusions

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 keywords. 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 [11] and its derivates) and are considering possible integrations with other tools for ontology building and management.

References


