Background Knowledge for Ontology Construction

Blaž Fortuna
Blaz.Fortuna@ijs.si
Institute Jožef Stefan
Ljubljana, Slovenia

Marko Grobelnik
Marko.Grobelnik@ijs.si
Institute Jožef Stefan
Ljubljana, Slovenia

Dunja Mladenič
Dunja.Mladenic@ijs.si
Institute Jožef Stefan
Ljubljana, Slovenia

ABSTRACT

In this paper we describe a solution for incorporating background knowledge into the OntoGen system for semi-automatic ontology construction. This makes it easier for different users to construct different and more personalized ontologies for the same domain. To achieve this we introduce a word weighting schema to be used in the document representation. The weighting schema is learned based on the background knowledge provided by user. It is than used by OntoGen's machine learning and text mining algorithms.

Categories and Subject Descriptors

H.3.7 [Digital Libraries]: *User issues*

General Terms

Algorithms, Human Factors.

Keywords

Semi-automatic Ontology construction, Background knowledge

1. INTRODUCTION

When using ontology-based techniques for knowledge management it is important for the ontology to capture the domain knowledge in a proper way. Very often different tasks and users require the knowledge to be encoded into ontology in different ways, depending on the task. For instance, the same document-database in a company may be viewed differently by marketing, management, and technical staff. Therefore it is crucial to develop techniques for incorporating user's background knowledge into ontologies.

In [4] we introduced a system called OntoGen for semi-automatic construction of topic ontologies. Topic ontology consists of a set of topics (or concepts) and a set of relations between the topics which best describe the data. The OntoGen system helps the user by discovering possible concepts and relations between them within the data.

In this paper we propose a method which extends OntoGen system so that the user can supervise the methods for concept discovery by providing background knowledge – his specific view on the data used by the text mining algorithms in the system.

To encode the background knowledge we require from the user to group documents into categories. These categories do not need to describe the data in details, the important thing is that they show to the system the user's view of the data – which documents are similar and which are different from the user's perspective. The process of manually marking the documents with categories is time consuming but can be significantly speeded up by the use of active learning [5, 7]. Another source of such labeled data could be popular online tagging services (e.g *Del.icio.us*) which allow the user to label the websites of his interests with labels he chose.

This paper is organized as follows. In Section 2 we introduce OntoGen system and in Section 3 we derive the algorithm for calculating word weights. We conclude the paper with some preliminary results in Section 4.

2. ONTOGEN

Important parts of OntoGen [4] are methods for discovering concepts from a collection of documents. For the representation of documents we use the well established bag-of-words representation, where each document is encoded as a vector of term frequencies and the similarity of a pair of documents is calculated by the number and the weights of the words that these two documents share. This method heavily relies on the weights associated with the words – the higher the weight of a specific word is the more probable it is that two documents are similar if they share this word. The weights of the words are commonly calculated by so called TFIDF weighting [8]. We argue that this provides just one of the possible views on the data and propose an alternative word weighting that also takes into account the background knowledge which provides the user's view on the documents.

OntoGen discovers concepts using Latent Semantic Indexing (LSI) [3] and k-means clustering [6]. The LSI is a method for linear dimensionality reduction by learning an optimal sub-basis for approximating documents' bag-of-words vectors. The sub-basis vectors are treated as concepts. The k-means method discovers concepts by clustering the documents' bag-of-words vectors into *k* clusters where each cluster is treated as a concept.

3. WORD WEIGHTING

3.1 Bag-of-Words and Cosine Similarity

The most commonly used representation of the documents in text mining is *bag-of-words* representation [5]. Let $V=\{w_1,\ldots,w_n\}$ be vocabulary of words. Let TF_k be the number of occurrences of the word w_k in the document. In the bag-of-words representation a single document is encoded as a vector $\boldsymbol{x}$ with elements corresponding to the words from a vocabulary providing some word weight, eg. $x^k = TF_k$.

Measure usually used to compare text documents is *cosine similarity* [5] and is defined to be the cosine of the angle between two documents' bag-of-words vectors,

$$sim(x_i,x_j)=\sum_{k=1}^{n} x_i^k \cdot x_j^k \Bigg/ \sqrt{\sum_{k=1}^{n} x_i^k \cdot x_i^k}\sqrt{\sum_{k=1}^{n} x_j^k \cdot x_j^k} .$$

Performance of both bag-of-words representation and cosine similarity can be significantly influenced by word weights. Each word from vocabulary V is assigned a weight and elements of vectors $\boldsymbol{x}_i$ are multiplied by the corresponding weights.

WWW 2006, May 23–26, 2006, Edinburgh, Scotland.
ACM 1-59593-323-9/06/0005.

3.2 SVM Feature Selection

Feature selection methods based on Support Vector Machine (SVM) [2] has been found to increase the performance of classification by discovering which words are important for determining the correct category of a document [1]. The method proceeds as follow. First linear SVM classifier is trained using all the features. Classification of a document is done by multiplying the document's bag-of-words vector with the normal vector computed by SVM,

$$\boldsymbol{x}^T\boldsymbol{w} = x^1w^1 + x^2w^2 + \ldots + x^nw^n,$$

and if the result is above some threshold b then the document is considered positive. This process can also be seen as voting where each word is assigned a vote weight w^i and when document is being classified each word from the document issues x^iw^i as its vote. All the votes are summed together to obtain the classification. A vote can be positive (the document belongs to the category) or negative (does not belong to the category).

A simple way of selecting the most important words for the given category would be to select the words with the highest vote values w^i for the category. It turns out that it is more stable to select the words with the highest vote x^iw^i averaged over all the positive documents. The votes w^i could also be interpreted as word weights since they are higher for the words which better separate the documents according to the given categories.

3.3 Word Weighting with SVM

The algorithm we developed for assigning weights using SVM feature selection method is the following:

1. Calculate a classifier for each category from the document collection (one-vs-all method for multi-class classification). TFIDF weighting schema can be used at this stage. Result is a set of SVM normal vectors $W = \{\boldsymbol{w}_j\, ;\, j=1,\ldots,m\}$, one for each category.
2. Calculate weighting for each of the categories from its classifier weight vector. Weights are calculated by averaging votes x^iw^i across all the documents from the category. Only weights with positive average are kept while the negative ones are set to zero. This results in a separate set of word weights for each category. By μ^j_k we denote weight for the k-th word and j-th category.
3. Weighted bag-of-words vectors are calculated for each document. Let $C(d_i)$ be a set of categories of a document d_i. Elements of vector $\boldsymbol{x}_i$ are calculated in the following way:

$$x_i^k = \left(\sum_{j \in C(d_i)} \mu_k^j\right) \cdot TF_k \quad .$$

This approach has another strong point. Weights are not only selected so that similarities correspond to the categories given by the user but they also depend on the context. Let us illustrate this on a sample document which contains words "machine learning". If the document would belong to category "learning" then the word "learning" would have high weight and the word "machine" low weight. However, if the same document would belong to category "machine learning", then most probably both words would be found important by SVM.

4. PRELIMENARY RESULTS

As a document collection for testing the above methods we chose Reuters RCV1 [9] dataset. We chose it because each news article from the dataset has two different types of categories (1) the topics covered and (2) the countries involved in it. We used a subset of 5000 randomly chosen documents for the experiments.

In the Figure 1 are the top 3 concepts discovered with k-means algorithm for both word weighting schemas. Documents are placed also in different concepts. For example, having two documents talking about the stock prices, one at the New York stock-exchange and the other at the UK stock-exchange. The New York document was placed in (1) *Market* concept (the same as the UK document) and in (2) *USA* concept (while the UK document was placed in (2) *Europe* concept).

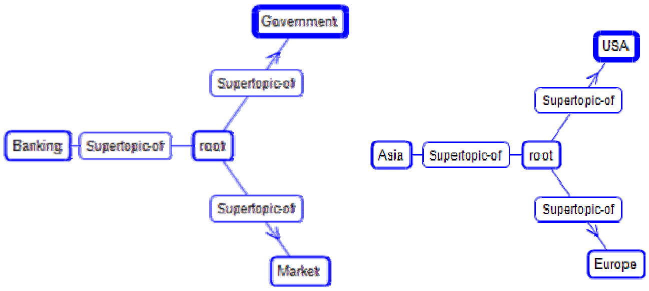

Figure 1. The top 3 discovered concepts for topic labels (left) and for country labels (right).

5. ACKNOWLEDGMENTS

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.

6. REFERENCES

[1] Brank J., Grobelnik M., Milic-Frayling N. & Mladenic D. Feature selection using support vector machines. Proc. of the Third International Conference on Data Mining Methods and Databases for Engineering, Finance, and Other Fields, 2002.

[2] Cristianini N. & Shawe-Taylor, J., An introduction to support vector machines, Cambridge University Press

[3] Deerwester S., Dumais S., Furnas G., Landuer T. & Harshman R. Indexing by Latent Semantic Analysis, J. of the American Society of Information Science, vol. 41/6, 391-407

[4] Fortuna B., Mladenic D. & Grobelnik M. Semi-automatic construction of topic ontology. Proc. of ECML/PKDD Workshop KDO 2005.

[5] Grobelnik M. & Mladenic D. Automated knowledge discovery in advanced knowledge management. J. of. Knowledge management 2005, Vol. 9, 132-149.

[6] Jain, A. K., Murty M. N., & Flynn P. J. Data Clustering: A Review, ACM Computing Surveys, vol 31/3, 264-323, 1999.

[7] Novak B., Mladenic D. & Grobelnik M. Text classification with active learning. Proceedings of GfKl 2005.

[8] Salton, G. Developments in Automatic Text Retrieval. Science, Vol 253, 974-979

[9] Lewis D., Yang Y., Rose T. & Li, F. RCV1: A New Benchmark Collection for Text Categorization Research. Journal of Machine Learning Research, 5:361-397, 2004.