Background knowledge for ontology construction

Blaz Fortuna and **Marko Grobelnik** and **Dunja Mladenic**[1]

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.

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], [8]. 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.

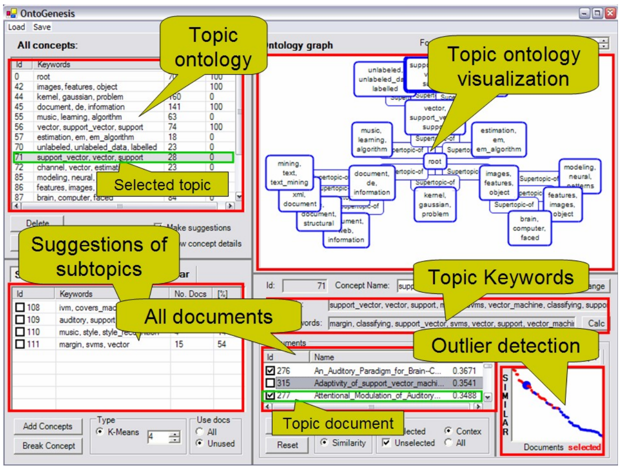
2 ONTOGEN

OntoGen [4] is a system for semi-automatic ontology construction, screenshot of the tool is presented in the Figure 1. Important part of OntoGen are methods for discovering concepts from a collection of documents. For the representation of the documents we use the well established bag-of-words representation which heavily relies on the weights associated with the words. The weights of the words are commonly calculated by so called TFIDF weighting. We argue that this provides just one of the possible views on the data and propose an alternative word weighting that 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 which approximates 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.

Both methods heavily rely on the representation of the documents. Namely, the document representation provides the vectors of the documents which LSI tries to approximate and, the basis for clustering algorithm is the similarity of document which also depends on the document representation.

By incorporating background knowledge directly into the document representation via word weighting, reflecting similarity between the documents, we enable our methods to discover concepts which resemble the view that the user has on the data.



OntoGenesis

All concepts:

Id	Keywords		
0	root	70	100
42	images, features, object		100
44	kernel, gaussian, problem	160	0
45	document, de, information	141	100
55	music, learning, algorithm	63	0
56	vector, support_vector, support	74	100
57	estimation, em, em_algorithm	18	0
70	unlabeled, unlabeled_data, labelled	23	0
71	support_vector, vector, support	28	0
72	channel, vector, estimati	23	0
85	modeling, neural,		
86	features, images,		
87	brain, computer, faced	84	0

Id	Keywords	No. Docs	[%]
□ 108	ivm, covers_mac		
□ 109	auditory, support		
□ 110	music, style, style_reco		
□ 111	margin, svms, vector	15	54

Id: 71 Concept Name: supp

support_vector, vector, support, m ...vms, vector_machine, classifying, suppo

...words: margin, classifying, support_vector, svms, vector, support, vector_machi

Id	Name	
☑ 276	An_Auditory_Paradigm_for_Brain-C...	0.3671
□ 315	Adaptivity_of_support_vector_machi...	0.3541
☑ 277	Attentional_Modulation_of_Auditory...	0.3488

Type: ● K-Means 4

Use docs: ○ All ● Unused

☐ Make suggestions

...lected ● Contex

● Similarity ☑ Unselected ○ All

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

[1] Institute Jozef Stefan, Slovenia, email: {blaz.fortuna, marko.grobelnik, dunja.mladenic}@ijs.si

3 WORD WEIGHTING

3.1 Bag-of-Words and Cosine Similarity

Most commonly used representation of the documents in text mining is bag-of-words representation. 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 x with elements corresponding to the words from a vocabulary, eg. $x^k = TF_k$. These vectors are in general very sparse since the number of different words that appear in the whole collection is usually much larger than the number of different words that appear inside one specific document.

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

$$sim(\mathbf{x}_i, \mathbf{x}_j) = \frac{\sum_{k=1}^{n} \mathbf{x}_i^k \mathbf{x}_j^k}{\sqrt{\sum_{k=1}^{n} \mathbf{x}_i^k \mathbf{x}_i^k} \sqrt{\sum_{k=1}^{n} \mathbf{x}_i^k \mathbf{x}_i^k}}. \qquad (1)$$

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

As we already mentioned, our approach is based on the word weights being the key to viewing the same data from different angels. We can use the weights to store the background knowledge since the weights define which words are important.

3.2 TFIDF

Most of the research on word weighting schemas was traditionally done in the information retrieval community. A typical goal in information retrieval is to find the most relevant document from the document collection for a given query. Many popular methods from information retrieval are based on measuring cosine similarity between the documents and a query and their performance can be significantly improved by appropriate weighting of the words.

Most of the popular methods for this task developed in last decades do not involve learning. Word weights are calculated by predefined formulas from some basic statistics of the word frequencies inside the document and inside the whole document collection [10]. These methods are base on intuition and experimental validation.

The most widely used is the TFIDF weighting schema [10] which defines elements of bag-of-words vectors with the following formula:

$$\mathbf{x}_i^k = TF_k \cdot \log(N \cdot IDF_k). \qquad (2)$$

The intuition behind this weighting schema is that the words which occur very often are not so important for determining if a pair of documents is similar while a not so frequent words occurring in the both documents is a strong sign of similarity. The TFIDF weighting can be easily modified to include category information by replacing IDF and number of documents with ICF and number of categories.

There are many extensions of this schema most famous being Okapi weighting schema [9] which we will skip here since it does not incorporate category information.

3.3 SVM Feature Selection

As we will see in the next chapter a different approach can also be taken for generating word weights based on feature selection methods. 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,

$$x^T w = x^1 w^1 + x^2 w^2 + \ldots + x^n w^n, \qquad (3)$$

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 $x^i w^i$ as its vote. All the votes are summed together to obtain the classification. A vote can be positive (document should belong to the category) or negative (the document should not belong to the category).

A simple and naive way of selecting the most important words for the given category would be to select the words with the highest vote values wi for the category. It turns out that it is more stable to select the words with the highest vote $x^i w^i$ 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.

3.4 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 $W = \{\mathbf{w}_j; j = 1, \ldots, m\}$, one for each category.
2. Calculate weighting for each of the categories from its classifier weight vector. Weights are calculated by averaging votes $x^i w^i$ 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_k^j 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 $\mathbf{x}_i$ are calculated in the following way:

$$\mathbf{x}_i^k = \left(\sum_{j \in C(d_i)} \mu_k^j \right) \cdot TF_k \ . \qquad (4)$$

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 PRELIMINARY RESULTS

4.1 Reuters RCV1 Dataset

As a document collection for testing our method we chose Reuters RCV1 [7] dataset. The reason for which we chose it is that each news article from the dataset has two different types of labels (categories). Each news article is assigned labels according to (1) the topics covered and (2) the countries involved in it. We used a subset of 5000 randomly chosen documents for the experiments.

A List with the 10 most frequent categories from the used subset of RCV1 dataset is shown in Table 1. The statistics are for the subset used in the experiments.

Table 1. List of 10 most frequent categories for topics and countries view.

TOPICS VIEW			COUNTRIES VIEW	
CCAT	corporate/industrial	46%	USA	33%
GCAT	government/social	30%	UK	11%
MCAT	markets	24%	Japan	6%
C15	performance	19%	Germany	4%
ECAT	economics	14%	France	4%
C151	accounts/earnings	10%	Australia	3%
M14	commodity/markets	10%	India	3%
C152	comment/forcast	9%	China	3%
GPOL	domestic politics	7%	EEC	3%
M13	money markets	7%	Hong Kong	2%

4.2 Results

In the Figure 2 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).

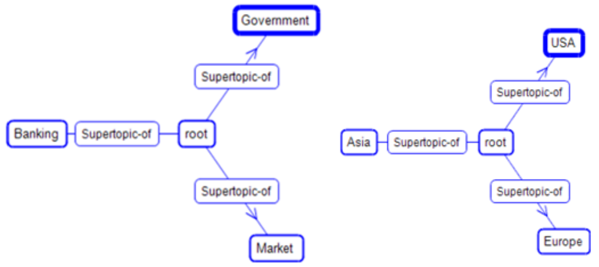


Figure 2. The top 3 discovered concepts for topic labels (left) and for country labels (right).

5 CONCLUSION

In this paper we have presented a method for learning document similarity measure trough selecting appropriate word weights for bag-of-words document representation model. We selected the word weights by training the SVM linear classifier for given categories and than extracting the word weights from the hyper plane normal vector. The learned word weighting schema was used to adjust the concept discovery methods in the OntoGen system to the user's domain knowledge.

As part of the future work we plan to extend this method to the text categorization task where category information is known only for the documents from training set.

ACKNOWLEDGEMENTS

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), NeOn Networked Ontologies (IST-2004- 27595) 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] N. Cristianini and J. Shawe-Taylor, An introduction to support vector machines, Cambridge University Press, 2000

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

[4] Fortuna, B., Mladenic, D., Grobelnik, M., (2005a). Semi-automatic construction of topic ontology. Proceedings of the ECML/PKDD Workshop on Knowledge Discovery for Ontologies.

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

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

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

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

[9] Robertson, S. E., S. Walker, M. M. Hancock-Beaulieu, M. Gatford and A. Payne. Okapi at TREC-4. The Fourth Text REtrieval Conference (TREC-4). 1996

[10] G.Salton. Developments in Automatic Text Retrieval. Science, Vol 253, pages 974-979. 1991

[11] Singhal, A., C. Buckley and M. Mitra. Pivoted Document Length Normalization. Proceedings of the 19th ACM SIGIR Conference on Research and Development in Information Retrieval. 1996