Comparing and Combining Two Approaches to Automated Subject Classification of Text

Koraljka Golub[1], Anders Ardö[1], Dunja Mladenić[2], and Marko Grobelnik[2]

[1] KnowLib Research Group, Dept. of Information Technology, Lund University, Sweden
{Koraljka.Golub, Anders.Ardo}@it.lth.se
[2] J. Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia
{Dunja.Mladenic, Marko.Grobelnik}@ijs.si

Abstract. A machine-learning and a string-matching approach to automated subject classification of text were compared, as to their performance, advantages and downsides. The former approach was based on an SVM algorithm, while the latter comprised string-matching between a controlled vocabulary and words in the text to be classified. Data collection consisted of a subset from Compendex, classified into six different classes. It was shown that SVM on average outperforms the string-matching approach: our hypothesis that SVM yields better recall and string-matching better precision was confirmed only on one of the classes. The two approaches being complementary, we investigated different combinations of the two based on combining their vocabularies. The results have shown that the original approaches, i.e. machine-learning approach without using background knowledge from the controlled vocabulary, and string-matching approach based on controlled vocabulary, outperform approaches in which combinations of automatically and manually obtained terms were used. Reasons for these results need further investigation, including a larger data collection and combining the two using predictions.

1 Methodology

The string-matching algorithm [1] classifies text documents into classes of the Ei classification scheme and thesaurus [2], based on simple string-matching between terms in the term list, derived from Ei, and terms in the document being classified. Ei contains several different types of terms and relationships, out of which we used captions, preferred and non-preferred terms. The term list (a model for classification) is organized as a list of triplets: term, class to which it maps, and weight. Each class in the original list is designated a number of term entries. No cut-off was used.

The second algorithm we used was linear SVM (support vector machine), a state-of-the art machine-learning algorithm, commonly used for text classification. We used binary SVM, the implementation from TextGarden [3]. We preprocessed the text by removing stop-words and representing each document using the standard bag-of-words approach containing individual words, enriched by frequent phrases (occurring at least four times in the data collection). The frequent phrases containing up to five consecutive words were automatically generated, as described in [4]. The model was trained on a part of data collection leaving the other part to be

J. Gonzalo et al. (Eds.): ECDL 2006, LNCS 4172, pp. 467 – 470, 2006.

classified using a standard approach of ten-fold cross validation. The binary classification model was automatically constructed for each of the six classes (see 2), taking all the training examples of the class as positive and all the other training examples as negative. Each example from the data collection was classified by each of the six models. For each example, we report all the classes that are above the threshold of zero.

2 Experimental Setting

Data collection consisted of a subset of paper records from the Compendex database [5], classified into six selected classes. Each document can belong to more than one class. Fields of the records that were used to classify are title, abstract and uncontrolled terms in the string-matching algorithm, and title and abstract in SVM.

In this first run of the experiment, only the six classes were selected in order to provide us with indications for further possibilities. Classes 723.1.1 (Computer Programming Languages), 723.4 (Artificial Intelligence), and 903.3 (Information Retrieval and Use) each had 4400 examples (the maximum allowed by the Compendex database provider), 722.3 (Data Communication Equipment and Techniques) 2800, 402 (Buildings and Towers) 4283, and 903 (Information Science) 3823 examples.

The linear SVM in the original setting was trained with no feature selection except the stop-word removal. Additionally, three experiments were conducted using feature selection, taking:

1. only the terms that are present in the controlled vocabulary;
2. the top 1000 terms from centroid tf-idf vectors for each class (terms that are characteristic for the class – descriptive terms);
3. the top 1000 terms from the SVM-normal trained on a binary classification problem for each class (terms that distinguish one class form the rest – distinctive terms).

In the experiments with string-matching algorithm, four different term lists were created, and we report performance for each of them:

1. the original one, based on the controlled vocabulary;
2. the one based on automatically extracted descriptive keywords from the documents belonging to their classes;
3. the one based on automatically extracted distinctive keywords from the documents belonging to their classes;
4. the one based on union of the first and the second list.

In lists 2, 3, and 4, the same number of keywords was assigned per class as in the original one.

Evaluation was based on comparing automatically assigned classes against the intellectually assigned classes given in the data collection. Precision (Prec.), recall (Rec.) and F1 measure were used as standard evaluation measures. Both standard

ways of calculating the average performance were used: macroaverage (macro.) and microaverage (micro.)

3 Experimental Results

We have experimentally compared performance of the two algorithms on our data in order to test two hypotheses both based on the observation that the two algorithms are complementary. Our first hypothesis was that, as the string-matching algorithm uses manually constructed model, we expect it to have higher precision than the machine learning with its automatically constructed model. On the other hand, while the machine-learning algorithm builds the model from the training data, we expect it to have higher recall in addition to being more flexible to changes in the data. Experiments have confirmed the hypothesis only on one of the six classes. Experimental results of the string-matching approach and the machine learning (SVM) approach (both using their original setting) are given in Table 1: SVM on average outperforms the string-matching algorithm. Different results were gained for different classes. The best results are for class 402, which we attribute to the fact that it has the highest number of term entries designating it (423). Class 903.3, on the other hand, has only 26 different term entries designating it in the string-matching term list, but string-matching largely outperforms SVM in precision (0.97 vs. 0.79). This is subject to further investigation.

Table 1. Experimental results comparing performance of the two approaches, and number of original terms per class (Terms). We can see that SVM performs better in all but one classes.

	String-matching (SM)				Machine learning (SVM)		
Class	**Terms**	**Rec.**	**Prec.**	**F1**	**Rec.**	**Prec.**	**F1**
402	423	0.58	0.49	0.53	**0.93**	**0.91**	0.92
722.3	292	0.12	0.26	0.16	**0.76**	**0.79**	0.78
723.1.1	137	0.34	0.32	0.33	**0.74**	**0.79**	0.76
723.4	61	0.37	0.39	0.38	**0.65**	**0.81**	0.72
903	58	0.28	0.61	0.38	**0.72**	**0.74**	0.73
903.3	26	0.32	**0.97**	0.48	**0.74**	0.79	0.76
Micro.		0.35	0.45	0.39	**0.78**	**0.81**	**0.78**
Macro.		0.34	0.51	0.38	**0.76**	**0.81**	**0.78**

The second hypothesis was that combining the two approaches via combining their vocabularies will result in improved performance. This hypothesis was not confirmed: both approaches have the best performance in the original setting (see Table 2). We attribute that to a large overlap between the controlled vocabulary and the document vocabulary that enables SVM to find the right terms for a good quality model.

Table 2. Macroavergaed experimental results comparing performance of SVM and string-matching approach (SM). We can see that both perform best using the original vocabularies.

				SVM – controlled		
	Rec.	**Prec.**	**F1**	**Rec.**	**Prec.**	**F1**
SVM – original (complete)	**0.76**	**0.81**	**0.78**	0.55	0.57	0.55
	SVM – descriptive			**SVM – distinctive**		
	Rec.	**Prec.**	**F1**	**Rec.**	**Prec.**	**F1**
Macroavg.	0.72	0.79	0.75	0.75	0.64	0.69
	SM - original (controlled)			**SM – distinctive + controlled**		
	Rec.	**Prec.**	**F1**	**Rec.**	**Prec.**	**F1**
Macroavg.	**0.55**	**0.68**	**0.61**	0.99	0.19	0.32
	SM– descriptive			**SM – distinctive**		
	Rec.	**Prec.**	**F1**	**Rec.**	**Prec.**	**F1**
Macroavg.	0.92	0.29	0.43	0.99	0.19	0.32

From Table 3 we can see that the string-matching algorithm the performance decreases due to a large drop in precision. Actually, almost every document gets all the six classes assigned, which increases recall to almost 100%. There is a possibility that low precision could be improved by introducing a cut-off value.

Acknowledgements

This work was supported by the Slovenian Research Agency and the IST Programme of the European Community under ALVIS (IST-1-002068-STP).

References

1. Golub, K. 2006. Automated subject classification of textual Web pages, based on a controlled vocabulary: challenges and recommendations. New review of hypermedia and multimedia, Special issue on knowledge organization systems and services, 2006(1).
2. Ei thesaurus, edited by J. Milstead, Engineering Information, Castle Point on the Hudson Hoboken, 1995. 2nd ed.
3. Grobelnik, M., Mladenic, D. Text Mining Recipes, Springer-Verlag, Berlin; Heidelberg; New York, 2006, accompanying software available at http://www.textmining.net.
4. Mladenic, D., Grobelnik, M. Feature selection on hierarchy of web documents. Journal of Decision Support Systems, 35, 45-87, 2003.
5. Compendex database. http://www.engineeringvillage2.org/.