Application of Lemmatization and Summarization Methods in Topic Identification Module for Large Scale Language Modeling Data Filtering

Lucie Skorkovská

University of West Bohemia, Faculty of Applied Sciences, Dept. of Cybernetics
Univerzitní 8, 306 14 Plzeň, Czech Republic
lskorkov@kky.zcu.cz

Abstract. The paper presents experiments with the topic identification module which is a part of a complex system for acquisition and storing large volumes of text data. The topic identification module processes each acquired data item and assigns it topics from a defined topic hierarchy. The topic hierarchy is quite extensive – it contains about 450 topics and topic categories. It can easily happen that for some narrowly focused topic there is not enough data for the topic identification training. Lemmatization is shown to improve the results when dealing with sparse data in the area of information retrieval, therefore the effects of lemmatization on topic identification results is studied in the paper. On the other hand, since the system is used for processing large amounts of data, a summarization method was implemented and the effect of using only the summary of an article on the topic identification accuracy is studied.

Keywords: topic identification, lemmatization, summarization, language modeling.

1 Introduction

In order to robustly estimate the parameters of statistical language models for natural language processing (automatic speech recognition, machine translation, etc.) the extensive amount of training data is required. It has been shown that not only the size of the training data is important, but also the right scope of the language models training texts is needed [9].

A complex system for acquisition and storing large volumes of text data was implemented [12] and one of its parts is a topic identification module used for large scale language modeling data filtering [11]. The module uses a language modeling based approach for the implementation of the topic identification.

Lemmatization has been shown to improve the results when dealing with sparse data in the area of information retrieval [4] and spoken term detection [10] in highly inflected languages, therefore the effects of lemmatization on topic identification accuracy is studied in the paper. On the other hand, since the system is used for processing large amounts of data, a summarization method was implemented and the effect of using only the summary of an article on the topic identification accuracy is studied.

P. Sojka et al. (Eds.): TSD 2012, LNCS 7499, pp. 191–198, 2012.

2 System for Acquisition and Storing Data

The topic identification module is a part of a system designed for collecting a large text corpus from Internet news servers described in [12]. The system consists of a SQL database and a set of text processing algorithms which use the database as a data storage for the whole system. One of the important features of the system is its modularity – new algorithms can be easily added as modules.

For the topic identification experiments the most important parts of the system are the text preprocessing modules. Each new article is obtained as a HTML page, then the *cleaning* algorithm is applied – it extracts the text and the metadata of the article. Then the *tokenization* and *text normalization* algorithms are applied – text is divided into a sequence of tokens and the non-orthographical symbols (mainly numbers) are substituted with a corresponding full-length form. The tokens of a normalized text are processed with a *vocabulary-based substitution* algorithm. Large vocabularies prepared by experts are used to fix the common typos, replace sequences of tokens with a multiword or to unify the written form of common terms. *Decapitalization* is also performed - substitutes the capitalized words at the beginning of sentences with the corresponding lower-case variants. The result of each of the preprocessing algorithm is stored as a text record in the database.

For the experiments described in this paper two new modules was added – automatic lemmatization module and automatic text summarization module.

2.1 Lemmatization Module

The task of the automatic lemmatization is to find a basic word form or a "lexical headword" of a given word. The use of some lemmatization preprocessing has been shown to be especially important for the highly inflected languages in the various tasks of natural language processing, such as keyword spotting or information retrieval.

The "lexical headword" of a word can be any of its forms, usually for example for a noun its singular nominativ is used or for a verb its infinitive. The particular base form of a word is given by creating the assignment between the word and its base form. The use of lemmatization leads to the reduction of the number of processed words – different forms of a word can be treated as one (its base form) – so it is especially advantageous in highly inflected languages, such as Czech, where for example a verb can have as much as fifty different forms.

The lemmatization module uses a lemmatizer described in the work [5]. This lemmatizer is automatically created from the data containing the pairs full word form – base word form. Based on this data a set of lemmatization rules and a vocabulary of base word forms is created. Also, a set of lemmatization examples is extracted, which is used for the lemmatization of the words that are not contained in the base word forms vocabulary. A lemmatizer created in this way has been shown to be fully sufficient in the task of information retrieval [6].

2.2 Summarization Module

Automatic text summarization is a process of creating a compression of a given text (or texts) with the preservation of as much information as is possible. The content of the information can be either generic or topic related. Summarization can be done in two ways – *summary by extraction*, where the text of the summary is extracted from the original text, or *summary by abstraction*, where the text of the summary is automatically generated to rephrase the contained information.

The task of the extractive summarization is to choose a subset of sentences with the maximal information content. Statistic algorithms for extractive text summarization often requires large training data – for example the work [7] uses a Bayesian classifier trained on a large corpus of professionally created abstracts to chose the sentences to include in the summary. On the other hand, when such training data is not accessible, the selection of the sentences is based on some heuristic features such as word frequency [8], position of sentences [2] in the text or the relation between sentences.

For the automatic summarization module an extractive generic summarization was chosen, as we want our summaries to preserve all the information contained in the original text, so the topic identification module can assign the correct topics. The implemented summarization algorithm selects the most important sentences in a text, where an importance of a sentence is measured by the importance of its words. One of the most commonly used measure for assessing the word importance in information retrieval area is the TF-IDF[1] measure, so we have decided to use it as well. The summary is created in a following way:

1. Split text to sentences and sentences to words.
2. For each term t in the document compute an *idf* weight:

$$idf_t = \log \frac{N}{N_t} \tag{1}$$

   where N is the total number of sentences in the document and N_t is the number of sentences containing the term t.
3. For each sentence s compute a term frequency $tf_{t,s}$ for each term. We have used the normalization of the term frequency by the maximum term frequency in the sentence.
4. The importance score S of each sentence in the document is computed as:

$$S_s = \sum_{t \in s} tf_{t,s} \cdot idf_t \tag{2}$$

5. The five sentences with the highest score S are included in the summary.

2.3 Topic Identification Module

The main purpose of the topic identification module is to filter the huge amount of data according to their topics for the future use as the language modeling training data. Currently, the topic identification module uses a language modeling based classification algorithm and assigns 3 topics to each article. Topics are chosen from a hierarchical system – a “topic tree”. Further information about the topic identification module can be found in [11].

[1] Term Frequency – Inverse Document Frequency.

Topic Tree. The topic hierarchy was built in a form of a topic tree, it is based on our expert findings in topic distribution in the articles on the Czech favorite news servers like *ČeskéNoviny.cz* or *iDnes.cz*. At present the topic tree has 32 generic topic categories like `politics`, `schools` or `sports`, each of this main category has its subcategories with the "smallest" topics represented as leaves of this tree. The deepest path in the tree has a length of four nodes(`industry` - `energetics` - `energy` - `solar`), an example can be seen on Figure 1.

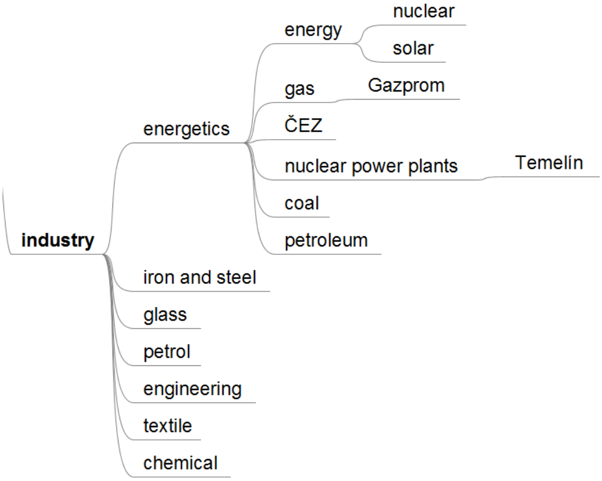

Fig. 1. Branch of the topic tree representing the industry topics

For the experiments the same topic tree as in [11] was used – it contains about 450 topics and topic categories, which correspond to the keywords assigned to the articles on the mentioned news servers. The articles with these "originally" assigned topics were used as training texts for identification algorithms.

Identification Algorithm. Current version of the topic identification module uses a language modeling based approach chosen due to the results of experiments published in [11]. This approach is similar to the Naive Bayes classifier, used for example in the work [1].

The probability $P(T|A)$ of an article A belonging to a topic T is computed as

$$P(T|A) \propto P(T) \prod_{t \in A} P(t|T) \tag{3}$$

where $P(T)$ is the prior probability of a topic T and $P(t|T)$ is a conditional probability of a term t given the topic T. The probability is estimated by the maximum likelihood estimate as the relative frequency of the term t in the training articles belonging to the topic T:

$$\hat{P}(t|T) = \frac{tf_{t,T}}{N_T} \tag{4}$$

where $tf_{t,T}$ is the frequency of the term t in T and N_T is the total number of tokens in articles of the topic T.

The goal of this language modeling based approach is to find the most likely topics T of an article A - for each article the three topics with the highest probability $P(T|A)$ are chosen. The prior probability of the topic $\hat{P}(T)$ is not used.

3 Evaluation

For the experiments on the effect of the automatic lemmatization and summarization algorithms three smaller collections containing the articles from the news server *ČeskéNoviny.cz* was separated from the whole corpus. This collections contain 5,000, 10,000 and 31,419 articles, details about the exact number of articles and the separation to training / testing data is shown in Table 1. The articles from *ČeskéNoviny.cz* have included the originally assigned keywords from their authors, which were used as the training and reference topics.

Table 1. Number of articles in the test collections and the training / testing data parts

collection name	articles	training	test
5k	5,000	4,000	1,000
10k	10,000	8,000	2,000
30k	31,419	27,000	4,419

Evaluation from the point of view of information retrieval (IR) was performed on the collections, where each newly downloaded article is considered as a query in IR and precision (P) and recall (R) is computed for the answer topic set:

$$P = \frac{T_C}{T_A}, \qquad R = \frac{T_C}{T_R} \tag{5}$$

where T_A is the number of topics assigned to the article, T_C is the number of correctly assigned topics and T_R is the number of relevant reference topics. An average of these measures is then computed across a set of testing articles. The F_1-measure is then computed from the (P) and (R) measures:

$$F_1 = 2\frac{P \cdot R}{P + R} \tag{6}$$

The training of the topic identification is done by counting the statistics containing the number of occurrences of each word in the whole collection, number of occurrences of each word in each document and the number of occurrences of each word in the documents belonging to a topic. These statistics can be trained from each kind of a text record in the database (result of a preprocessing step). For our experiments, the statistics for each of the collections were trained from the text preprocessed by following modules:

– *Replaced* - tokenization, text normalization, decapitalization and vocabulary-based substitution modules

- *Lemma* - tokenization, text normalization and lemmatization modules
- *Summary* - tokenization, text normalization, decapitalization, vocabulary-based substitution and summarization modules

The results of the topic identification on the different sized and preprocessed collections can be seen in Table 2. Testing articles were preprocessed in the same way as the collections – *replaced*, *lemma* and *summary* (first 3 columns of the table). For the summarization testing, there was done also the summary from the lemmatized testing articles (for the combination with *lemma* preprocessed collection – column 4 of the table) and from the *replaced* testing articles (combination with *replaced* preprocessed collection – column 5 of the table).

Table 2. Results of topic identification on different collections

coll./art.		**replaced**	**lemma**	**summary**	**lemma/summary**	**replaced/summary**
5k	P	0.5366	**0.5547**	0.5028	*0.5457*	0.5374
	R	0.5544	**0.5754**	0.5155	*0.5686*	0.5546
	F_1	0.5454	**0.5649**	0.5091	*0.5569*	0.5459
10k	P	0.5481	**0.5536**	0.5024	*0.5378*	0.5293
	R	0.5472	**0.5555**	0.4979	*0.5421*	0.53
	F_1	0.5476	**0.5546**	0.5002	*0.54*	0.5296
30k	P	0.5864	**0.5859**	0.5387	*0.5588*	0.5598
	R	0.6125	**0.6155**	0.5616	*0.5921*	0.5884
	F_1	0.5992	**0.6003**	0.5499	*0.575*	0.5737

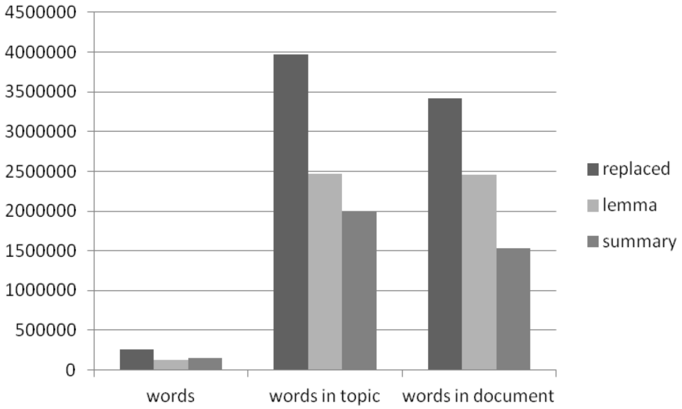

Fig. 2. Comparison of the number of lines of the database tables used to store the topic identification statistics for the 3 types of the training articles preprocessing

From the table we can draw following conclusions:

- The summarized text is not suitable for training topic identification statistics, results in column summary are the worst for all sizes of collections. This is not surprising, as much less text is used for counting the statistics so the topic important words may be missing.

- The use of lemmatization seems to improve the topic identification results, especially on the smaller collections 5k and 10k. Lemmatization also reduces the size of the database tables used to store the topic identification statistics (for comparison see Figure 2).
- The most interesting finding of our experiments can be seen in column lemma/summary. When needed, a faster computation of topic identification using summarized and lemmatized texts can be used with a minimum loss on the topic identification accuracy.

4 Conclusions and Future Work

The evaluation of topic identification accuracy suggests that the lemmatization algorithm should be used for text preprocessing, as the accuracy of topic identification are slightly better than without the lemmatization. On the top of that, is was shown that the use of lemmatization reduces the size of stored database word statistics tables almost about a half. The lemmatized database with the combination of summarized test articles has only slightly worse topic identification accuracy, but the time needed for the topic identification of an article is reduced as the computation of the probability $P(T|A)$ of an article belonging to a topic is done over a reduced set of words.

For the future work, we would like to implement more complex summarization algorithm, currently the Graph-based LexRank method [3] for text summarization is being implemented. It may have an effect on the topic identification accuracy if the sentences in the summary will be chosen better. Also, we would like to use the summarization module for example for the multi-document topic oriented summaries or for the actuality summaries – summary of what happened in the last week for example.

Acknowledgments. This research was supported by the Ministry of Education, Youth and Sports of the Czech Republic project No. LM2010013 and by the grant of the University of West Bohemia, project No. SGS-2010-054.

References

1. Asy'arie, A.D., Pribadi, A.W.: Automatic news articles classification in indonesian language by using naive bayes classifier method. In: Proceedings of the 11th International Conference on Information Integration and Web-based Applications & Services, iiWAS 2009, pp. 658–662. ACM, New York (2009)
2. Edmundson, H.P.: New methods in automatic extracting. J. ACM 16(2), 264–285 (1969)
3. Erkan, G., Radev, D.R.: Lexrank: graph-based lexical centrality as salience in text summarization. J. Artif. Int. Res. 22(1), 457–479 (2004)
4. Ircing, P., Müller, L.: Benefit of Proper Language Processing for Czech Speech Retrieval in the CL-SR Task at CLEF 2006. In: Peters, C., Clough, P., Gey, F.C., Karlgren, J., Magnini, B., Oard, D.W., de Rijke, M., Stempfhuber, M. (eds.) CLEF 2006. LNCS, vol. 4730, pp. 759–765. Springer, Heidelberg (2007)

5. Kanis, J., Müller, L.: Automatic Lemmatizer Construction with Focus on OOV Words Lemmatization. In: Matoušek, V., Mautner, P., Pavelka, T. (eds.) TSD 2005. LNCS (LNAI), vol. 3658, pp. 132–139. Springer, Heidelberg (2005)
6. Kanis, J., Skorkovská, L.: Comparison of Different Lemmatization Approaches through the Means of Information Retrieval Performance. In: Sojka, P., Horák, A., Kopeček, I., Pala, K. (eds.) TSD 2010. LNCS, vol. 6231, pp. 93–100. Springer, Heidelberg (2010)
7. Kupiec, J., Pedersen, J., Chen, F.: A trainable document summarizer. In: Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 1995, pp. 68–73. ACM, New York (1995)
8. Luhn, H.P.: The automatic creation of literature abstracts. IBM J. Res. Dev. 2(2), 159–165 (1958)
9. Psutka, J., Ircing, P., Psutka, J.V., Radová, V., Byrne, W., Hajič, J., Mírovský, J., Gustman, S.: Large vocabulary ASR for spontaneous Czech in the MALACH project. In: Proceedings of Eurospeech 2003, Geneva, pp. 1821–1824 (2003)
10. Psutka, J., Švec, J., Psutka, J.V., Vaněk, J., Pražák, A., Šmídl, L., Ircing, P.: System for fast lexical and phonetic spoken term detection in a czech cultural heritage archive. EURASIP J. Audio, Speech and Music Processing 2011 (2011)
11. Skorkovská, L., Ircing, P., Pražák, A., Lehečka, J.: Automatic Topic Identification for Large Scale Language Modeling Data Filtering. In: Habernal, I., Matoušek, V. (eds.) TSD 2011. LNCS, vol. 6836, pp. 64–71. Springer, Heidelberg (2011)
12. Švec, J., Hoidekr, J., Soutner, D., Vavruška, J.: Web Text Data Mining for Building Large Scale Language Modelling Corpus. In: Habernal, I., Matoušek, V. (eds.) TSD 2011. LNCS, vol. 6836, pp. 356–363. Springer, Heidelberg (2011)