News Article Classification Based on a Vector Representation Including Words’ Collocations

Michal Kompan and Mária Bieliková,

Slovak University of Technology, Faculty of Informatics and Information Technologies,
Ilkovičova 3, 842 16 Bratislava, Slovakia
{kompan, bielik}@fiit.stuba.sk

Abstract. In this paper we present a proposal including collocations into the pre-processing of the text mining, which we use for the fast news article recommendation and experiments based on real data from the biggest Slovak newspaper. The news article section can be predicted based on several article’s characteristics as article name, content, keywords etc. We provided experiments aimed at comparison of several approaches and algorithms including expressive vector representation, with considering most popular words collocations obtained from Slovak National Corpus.

Keywords: text pre-processing, news recommendation, news classification, vector representation

1 Introduction and related work

Nowadays no one is discussing the need for the web personalization. One of the ways in which personalization is performed represents recommendations. Recommendation task can be defined as follows:

$$\forall c \in C, s_c^{'} = argmax_{s \in S} u(c, s)$$

where C represent users, S represents objects of recommendation and u is the usefulness function (usefulness of an object for specific user).

Recommender systems had become important part of well-known web portals in several domains as online shops, libraries or news portals for years. News portals are characteristic with thousands of daily added articles with high information decrease degree, so the one from most relevant recommender systems’ attributes are reaction time of processing and the start of recommending new articles.

There are two widely used approaches to the recommendation: collaborative filtering based on an assumption that interest is shared between similar users and the content based recommendation where computed “associations” between entities (generally similarity relation) based on extracted useful information on the entity content are used. We focus on the content based recommendation in news domain.

Several recommender systems in the news domain (OTS, NewsMe, Pure, Google News, NewsBrief [1]) have been proposed in the last decade. The collaborative recommender on SME.SK [10] and the content based recommenders TRecom [12] and Titler [7] have been proposed for the Slovak language.

1.1 Classification task

Content based recommendation obviously includes recommended entities classification aimed at finding relations between entities. Considering recommending news articles, we are often limited to a specific source of articles – news portal. In such a portal often every article has its own category (mostly assigned by a human), which is rich and reliable source of important data (in the context of recommendation and similarity search). Nowadays researchers focus on aggregating recommender systems like NewsBrief or Google News, where several news portals over the world are monitored and used for generating recommendations. One of the possible solutions for the aggregation news from several portals is the classification based on articles categories from various portals. Several classifiers for news articles have been proposed respectively [8].

Main goal of the document classification is to assign one or more (probability) categories of the classified document. In the literature the classification task is often divided into supervised and unsupervised methods [4], where an unsupervised method refers to the document clustering. Unsupervised methods are based on an assumption that documents having similar content should be grouped into the one cluster. Hierarchical clustering algorithms have been intensively studied as well as neural network models. Several methods are used as Support vector machines, Naïve Bayes or Neural networks [11], where Naïve Bayes outperforms the others [2].

1.2 Text representation for classification

Because of the high information value decrease and the high dynamics in the news domain, there is need to process objects of the recommendation (articles) in a fast manner. For this purpose several text representations have been proposed [5].

The simplest method for representation of an article is the Bag of Words (BoW). As the best unit for the text representation is a term [5], BoW consists of term from the text. Other often used method of text representation is Vector Space Model, which adds weights (term frequency) to terms from BoW. In this way we obtain vectors representing text documents. It is clear that these representations have a huge problem with dimensionality and thus with the performance of any information retrieval method applied on. Various enhancements have been proposed as binary representation, ontology models or N-Grams [5].

Some methods do not consider all terms extracted from texts, but only the relevant. For the keywords or relevant term extraction Latent Semantic Indexing is often used. When extracting terms from semi-structured documents (such as HTML) additional information is used as HTML tags [5] for the relevant terms recognition.

2 News article representation proposal

In the domain of news recommendation, the time complexity of the classification process is critical. To reduce the space of words and to extract relevant information from articles often a vector representation of text is used. This brings usual the words' space reduction and accuracy improvements (if information extraction is included). We have proposed a vector representation (Table 1) based on the extraction of important (distinctive) terms from the article, in order to reduce the dimension of the space of words.

Table 1. Vector representation for an article.

Vector part	Weights
Title	*transplantation_0.5* *face_0.5*
TF of title words in the content	*transplantation_0.0178571428571429* *face_0.0714285714285714*
Category	*Sme.sk_0.5* *PRESS_FOTO_1.0*
Keywords	*clinc_0.0357142857142857* *surface_0.0178571428571429* *nose_0.0178571428571429* *tooth_0.0178571428571429* *nerve_0.0178571428571429* *masculature_0.0178571428571429* *patient_0.0178571428571429* *scale_0.0178571428571429*
Names/Places	*Cleveland_1*
CLI	*0.2543*

The article vector consists of six parts:

- *Title* - Article vector comprises lemmatized words from article title. It consists of approximately 5 words (150 000 Slovak article dataset). We suggest that article title should be in most occurrences good describing attribute.
- *Term Frequency of title words in the content* - We used TF to estimate the article name confidence. If the article name is abstract and do not correspond to article content, we can easily discover this situation.
- Keywords – We store 10 most relevant keywords. News portals have a list of keywords for every article usually. These are unfortunately at different abstraction level over various portals thus we have our own keywords list, which is based on TF-IDF list calculated over the dataset (100 000 Slovak news articles SME.SK).
- *Category* - This category part is constructed based on the portal specific category hierarchy (optional), while the hierarchy is represented by a tree structure.
- *Names/Places* - In this step we extract list of names and places obtained from the article content - as words starting with upper letter and with no full-stop before (precision = 0.934, recall = 0.863).
- *CLI* - Coleman-Liau readability index provides information about the level of the understandability of the text.

Most of the vector parts do not depend on a particular news portal, and can be easily extracted from a standard article. The only one dependent part in our representation is the Category. If we want to abstract of this, it is necessary to find similar articles over various portals and then respectively create corresponding virtual categories (the text classification task).

2.1 Text pre-processing and collocations

The text pre-processing plays critical role in the text classification process. It can significantly reduce space of words, but on the other hand, it can easily decrease the information value. In our experiments we work with texts in Slovak language.

The article pre-processing can be divided into several steps [9]:

- *Tokenization* – A simple strategy is to just split the text on all non-alphanumeric characters. As far as this step is language depending, some information (special addresses, names etc.) can be lost. Thus, advanced techniques are need for text tokenization, considering local habits.
- *Dropping common terms: stop-words* – For every language we can easily identify most common words without any or only with small information value (and, is, be, in etc.) By removing these words we are able to significantly reduce the words' space while in the most cases the information value of processed texts remains.
- *Normalization* – In other words the process of creating equivalence classes of terms. The goal is to map words with the same sense to the one class (e.g. "USA" and "U.S.A.").
- *Stemming and lemmatization* – Documents contain different words' forms and there are families of related words with similar meanings (car, cars, car's, cars' - car). For the English language the most common stemmer is Porter Stemmer, for flexive languages such as Slovak language it is more complicated.

As a result of the pre-processing step (in connection to the needs of our vector representation) we obtain:

- lemmatized article title (without stop words and punctuation),
- 10 most relevant keywords, the list of Names and Places.

The most frequent words occurred in specific language together are considered as collocations (bigrams). In order to improve the pre-processing step and to increase the information gain, we introduce words collocations into the text pre-processing step. We expect that enhancing the pre-processing step with words collocations will lead to the article similarity or classification tasks improvement.

We extracted word collocations from the Slovak national corpus (Ľ. Štúr Institute of Linguistic, Slovak Academy of Sciences). The example of collocations for word "conference" in Slovak are "central", "OSN", "focused". The most frequent collocations in general are stop words or punctuations. We do not consider such words ("in the", "does not" etc.).

In other words we enhanced pre-processing step while not only stop words, but collocations are removed. This leads to word space reduction. Our hypothesis is that after removing the collocations the information value of the pre-processed text remains the same – the information gain (words with distinctive characteristics) will remain and classification task accuracy will not decrease.

3 Hypothesis and design of experiments

Our hypothesis is that introducing words' collocations (removing collocations in the pre-processing step) can improve classification task results over the dataset. Thus we suggest several experiments, with various initial settings and classification algorithms.

For the classification task experiments we use SME.SK dataset from the project SMEFIIT [3]. We have total of 1 387 articles from 20 categories (extracted directly from news portal) in our dataset. Each article consists of the title, the article content and the real section in which was assigned by the article author. For each article we constructed representative article vector as described in Section 2. For the implementation and experiments we used RapidMiner [6] as one of the well-known and widely used information discovery environment.

First, we investigated which one from weighting techniques performs best. For this purpose we modelled a standard classification task with Naïve Bayes, K-NN and Decision trees as a classificatory. Weights for words were step by step calculated as TF-IDF, Term frequencies, Term occurrences and Binary term occurrences. We also used a pruning method where all words with the weight below 3.0 or above 30.0 percent were pruned. The pruning has negative impact on the whole process computation complexity. However, this can be compensated by the proposed vector representation.

In the second experiment we investigated which of the classifiers perform best for the classification task. We considered K-nearest neighbour, Naïve Bayes and Decision trees and their implementations in RapidMiner. For these methods we also evaluated the best weight function and these best-performers were used in next experiments.

Our aim was to evaluate properties of proposed article representation for classification task. So we performed all the experiments for both classical (TF-IDF, Term Frequency, Term Occurrences, Binary Term Occurrences) and our proposed representations. We do not use whole vector representation - the Category part was excluded (it is used as a learner for supervised learning).

Similarly, all experiments were performed for standard pre-processing as we described in the section 2 (without collocation remove/add). As the next step we added words' collocations to both representations and we studied performance changes. In the next experiment collocation were removed instead of added. Because most frequent collocations are stop words or words with a small information value, we decided to pre-process these words' collocations too and not to include most frequent collocations.

4 Results of experimental evaluation and discussion

For each experiment we measured the classification accuracy (ration between correctly and incorrectly classified articles) for pruned and not pruned data respectively. The evaluation was performed as "X-validation" (x=10) with stratified sampling. In Table 2 we present the weight functions' comparison for three classifiers.

As we can see in almost all cases classification performs the best when pruning method was active. Only in the classification with Naïve Bayes and with the standard representation no pruning outperformed the pruned data. The difference between pruned and not pruned data is significant. It is important to note, that while not pruning do not brings better results as pruning in general it also takes almost 10x longer as a classification with pruning.

Table 2. Naïve Bayes, K-NN and Decision Trees classifiers comparison for various weight functions (TF-IDF, Term Frequency, Term Occurrences, Binary Term Occurrences) considering vector (Vec.) and standard (Std.) representations (Classification accuracy).

	Naïve Bayes				K-NN				Decision Trees			
	No pruning		Pruning		No pruning		Pruning		No pruning		Pruning	
	Vec.	Std.	Vec.	Std.	Vec.	Std.	Vec.	Std.	Vec.	Std.	Vec.	Std.
TF-IDF	*57,9*	*51,6*	*78,3*	*44,0*	*55,6*	*54,0*	*66,0*	*52,7*	*98,7*	*21,9*	*94,6*	*21,9*
TF	*64,9*	*52,2*	*81,1*	*45,5*	*61,2*	*52,8*	*72,1*	*50,2*	*98,4*	*21,4*	*94,5*	*20,1*
TO	*63,0*	*48,6*	*85,2*	*39,5*	*66,1*	*38,5*	*84,6*	*39,6*	*98,8*	*21,5*	*94,8*	*21,0*
BO	*72,8*	*45,3*	*89,0*	*38,3*	*72,9*	*22,2*	*88,9*	*19,8*	*98,7*	*21,9*	*94,8*	*21,9*

Our proposed vector representation significantly outperforms standard article representation in the classification accuracy (Fig. 1). We can say that our proposed representation extracts relevant information (words with distinctive characteristics) and can be used for various information retrieval tasks not only for the similarity computation.

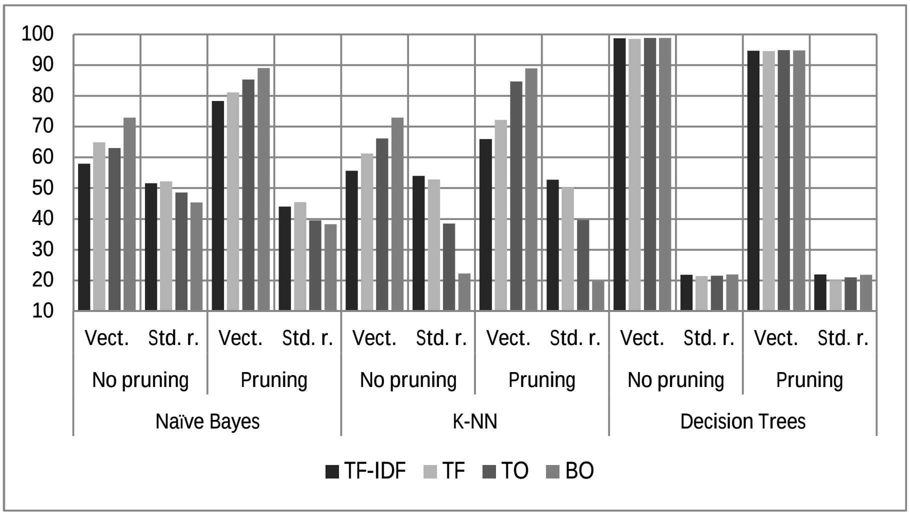

Fig. 1. Classifiers comparison (Classification accuracy) with various pre-processing methods (TF-IDF, Term Frequency, Term Occurrences, Binary Term Occurrences).

The highest accuracy increase, that we can observe, is using Decision Trees, which seems to be the best classifier for our task (Average improvement 75,22%). On the other hand classification with Decision trees takes the longest time even in case of vector representation and pruning included. Results when using Decision tree are "flat" in comparison to other approaches. This can be explained by the used approach, when the computed weights were not used.

We provided experiments for every possible combination of a weight function, the classifier, the representation and also considering words' collocations. Because of similar patterns we do not provide complete results for the collocations excluded. Aggregated results (mean of 4 weight functions) for the classification with collocations consideration can be seen in Fig. 2.

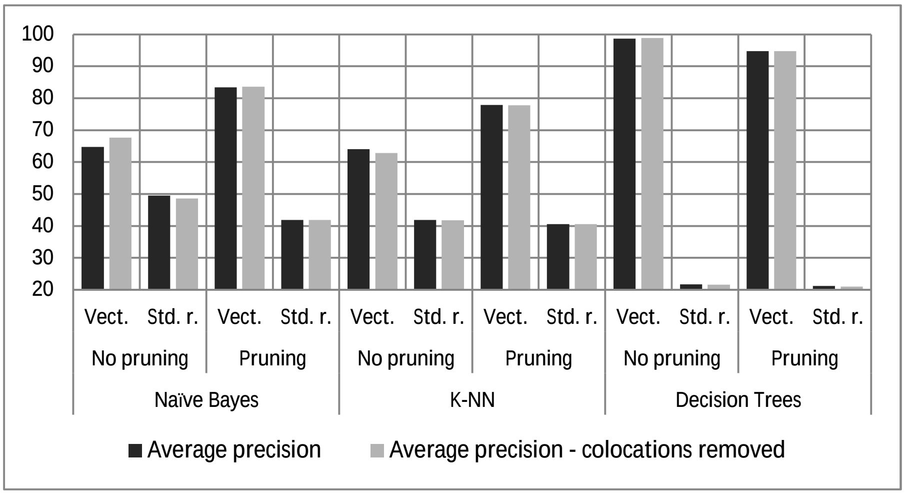

Fig. 2. Classifiers comparison with collocation consideration (average classification accuracy for 4 weight functions).

Our hypothesis appears to be wrong, i.e. excluding collocations in the pre-processing step does not significantly improve classification task. However, we can see, that removing collocations did not degrade the classification accuracy while it reduces words' space, in other words, correlated words were removed. For the similarity computation task it can be interesting to experiment with word collocations adding. In this case the sub-group without stop words and low information gain words should be carefully selected.

5 Conclusions

In this work we compared several classification methods applied to news articles considering proposed vector representation. The article classification allows us to abstract from concrete news portal and to start recommending and aggregating articles from various portals.

We enhanced pre-processing process by introducing words' collocations excluding. The proposed vector representation outperforms standard representation not only in the way of the classification accuracy (the best improvement *77,27%*) but it reduces the computation complexity of the classification process which is strictly connected to the computation time. Such a representation and category classification also are language independent. When we will replace collocations statistics and stop words list, we are able to use our proposed method for other languages.

Introducing collocations to the process of pre-processing does not bring improvement of pre-processing. On the other hand, collocations reduce the word space while do not decrease the information value at the same time.

Proposed vector representation can be used for content-based news recommendation and also to aggregate news articles from various news portals (using category classification) in a fast and effective way. As there are only few language depending steps during the pre-processing process, various languages can be included respectively.

References

1. Billsus D, Pazzani M. Adaptive news access. In Brusilovsky, P., Kobsa, A., Niejdl, W., eds.: The Adaptive Web: Methods and Strategies of Web Personalization. LNCS 4321. Springer-Verlag, Berlin Heidelberg New York. (2007).
2. Burgin, R. The retrieval effectiveness of five clustering algorithms as a function of indexing exhaustivity. Journal of the American Soc. for Inf. Science, Vol.46 (8), pp. 562-72, (1995).
3. Barla, M., Kompan, M., Suchal, J., Vojtek, P., Zeleník, D., Bieliková, M. News recommendation. In Proc. of the 9th Znalosti, Jindrichuv Hradec, pp. 171-174, (2010).
4. Göker, A., Davies, J. Inform. Ret.: searching in the 21st cent.. John Wiley & Sons, (2009).
5. Harish, B.,S., Guru, D., S., Manjunath, S. Representation and Classification of Text Documents: A Brief Review. IJCA,Special Issue on RTIPPR (2), pp.110–119, (2010).
6. Jungermann, F. Information Extraction with RapidMiner. In Proc. of the GSCL Symposium 'Sprachtechnologie und eHumanities' 2009. Duisburg, (2009).
7. Kompan, M., Bieliková, M. Content-Based News Recommendation. In Proc. of the 11th Conf. EC-WEB, Springer Verlag Berlin, pp. 61-72, (2010).
8. Kroha,, P., Baeza-Yates, R.A. A Case Study: News Classification Based on Term Frequency. In Proceedings of DEXA Workshops, pp. 428-432, (2005).
9. Manning, D., Raghavan, P., Schutze, H., Introduction to Information Retrieval, Cambridge University Press, (2008).
10. Suchal, J., Návrat, P. Full text search engine as scalable k-nearest neighbor recommendation system. AI 2010, IFIP AICT 331, Springer, pp. 165-173, (2010).
11. Witten, I.H., Frank, E. Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann Ser. in Data Manag. Sys.. Morgan Kaufmann, second edition, (2005).
12. Zeleník, D., Bieliková, M. News Recommending Based on Text Similarity and User Behaviour. In WEBIST '11: Proc. of the 7th Int. Conf. On Web Informations Systems and Technologies. Noordwijkerhout, Holland, 2011, to appear.