Ongoing Research in Document Classification at the „Lucian Blaga” University of Sibiu

Radu Cretulescu, Daniel Morariu, Lucian Vintan

"Lucian Blaga" University of Sibiu, Engineering Faculty, Computer Engineering Department

{radu.kretzulescu, daniel.morariu, lucian.vintan}@ulbsibiu.ro

Introduction

Most real-world information can be found in text documents. These data are considered to have a semi-structured format because they contain little meta-information about their structure. Modeling and implementation techniques for working with semi-structured data were constantly developed in recent years. Moreover, applications for information retrieval as methods for data indexing have been adapted to work with these unstructured documents.

Traditional information retrieval techniques become inadequate for searching large collections of unstructured or semi-structured data. Usually, only a small fraction from the available documents is relevant to the user at a time. Without knowing what these large data collections contain it is difficult to formulate effective queries for analysis and retrieval of "interesting", relevant and useful results to the user [6].

Machine Learning offers two approaches on how a "machine can learn" text documents, using supervised learning techniques (classification) and unsupervised learning techniques (clustering).

In our research we started from the premise to use purely computational methods to retrieve information from text documents. Although in some cases we tried to add a certain "amount of syntactic information" in various algorithms.

The overall aim of our work is to improve the performance of classification and clustering for text documents, through advanced supervised and unsupervised learning techniques.

To achieve this purpose we considered the following aspects:

- develop some meta-classifiers and determine solutions to improve the accuracy of their classification;
- improve the accuracy of classification for text documents developing hybrid meta-classifiers based on SVM, Bayes-type and Genetic Algorithms as selectors and a neural network in the post classification phase;

- demonstrate the utility for the representation of documents based on suffix trees (STDM - Suffix Tree Document Model) in some clustering algorithms;
- development and evaluation of new metrics to determine similarity between documents represented by the STDM model clustering algorithms;

Our team belonging to the ACAPS Research Centre –

http://acaps.ulbsibiu.ro/index.php/en/ had two major research directions in the text mining field: clustering and classification of text documents. We have used in our experiments two different data sets. The first one was extracted from the Reuters 2000 database. In this set after applying the preprocessing steps we have obtained a training set with 4702 documents and a testing set with 2351 pre-labeled documents with 1309 features. The documents belong to 16 classes. This set was used with our developed classification algorithms.

The second document set was extracted from RSS news feeds from the Reuters and BBC news agencies. The extracted documents were grouped in 7 subsets and used in clustering algorithms. These data were also pre-labeled for simplifying the evaluation process of the clustering.

Main research approaches for text documents classification

Given the fact that many new algorithms exploiting synergism of simple classification algorithms which can lead to good results in the automatic classification of text documents, we proposed a meta-classifier based on 8 SVM classifiers [7] and one Bayesian classifier. The purpose of the meta-classifier was to choose a winner class for a given document or to choose a classifier which had to classify a given document. Thus we developed 3 major types of meta-classifiers: non-adaptive, adaptive and hybrid meta-classifiers. For the experiments we have used the Reuters data set presented in the section above with 16 classes and 1309 features of the documents.

Non-adaptive meta-classifiers

In this section we present our developed non-adaptive meta-classifiers [1]. This meta-classifier contains 8 Support Vector Machine classifiers and one Naïve Bayes classifier. The output of each classifier is a vector with 16 scalars corresponding to the confidence degree given by the classifier for each class. This type of meta-classifier must choose a class for a given text document. Using first the majority vote (MV) method didn't bring acceptable results for the classification accuracy**.** We have observed that in some cases where the documents were misclassified the classes from the second position were the right ones. So we have decided to use all values returned by each classifier and sum all values generated by each simple classifier for each class separately. So we have obtained a vector with 16 scalars corresponding to the confidence given by the classifiers for each class.

The results of this simple meta-classifier are better than majority voting, but not significantly.

Another original approach was trying different values to weight the values of each class of vectors generated by the classifiers. These vectors' components were weighted, according to the order obtained by each class in the vector. The best obtained result generates 301 documents incorrectly classified, representing an 87.20% classification accuracy. This result was obtained when we have used a linear weighting step of "0.5".

In another set of experiments we have used genetic algorithms to solve the "Design Space Exploration" problem for finding the optimal weights that might be used in the meta-classifier presented above. This approach leads to a substantial improvement of classification accuracy. Thus, the accuracy of the meta-classifier with calculated weights have improved on average by 1.16% in the case of using "Roulette Wheel" selection of chromosomes, reaching an average of 88.36%. When we have used the Gauss Selection method for selecting the chromosomes from a population, the improvement was 1.17%, reaching 88.37% on average for the classification accuracy. The best result was obtained by this meta-classifier in an experiment using the Gauss selection method of chromosomes where the classification accuracy was 88.55% which is the best result for this type of meta-classifiers.

Adaptive meta-classifiers

In this section we present our developed adaptive meta-classifiers [1]. This type of meta-classifier must adaptively choose a classifier for classifying a given text document.

First we have used 8 SVM classifiers presented in [7]. For this meta-classifier we have computed the upper limit of the classification accuracy using non-adaptive aggregation methods which was 94.21%, because some documents from the testing set couldn't be classified correctly by any of the 8 included SVM classifiers. After including a special designed Bayesian classifier into the meta-classifier the upper limit raises to 98.63%.

We have designed a meta-classifier to learn the input data and we are expecting that the number of correctly classified samples will be greater than the number of incorrectly classified input samples. So that our meta-classifier will learn only the input samples incorrectly classified. As a consequence the meta-classifier will contain for each classifier a self-queue where are stored all incorrectly classified documents. Therefore, this meta-classifier contains 9 queues attached to the component classifiers.

Considering an input document (current sample) that needs to be classified, first we randomly chose one classifier. We compute the distance between the current sample and all samples that are in that self-queue of the selected classifier. If we obtain at least one distance smaller than a predefined threshold we renounce to use that selected classifier. In this case we randomly select another classifier. If there are cases when all component classifiers are rejected, however, we will choose that classifier with the greatest distance.

After selecting the classifier we use it to classify the current sample. If that selected classifier succeeds to correctly classify the current document, nothing is done. Otherwise, we will put the current document into the selected classifier's queue. We did this because we want to prevent that this component to further classify this kind of documents. To see if the document is correctly or incorrectly classified we compare the proposed class with Reuters proposed class that we considered to be perfect.

One document is written into the queue of misclassified documents only when the selected classifier proposes a different result than the result proposed by Reuters.

We have used 2 metrics – Euclidean distance and cosine distance - for computing the distance between the documents.

The results for the meta-classifiers using the selection based on Euclidean distance (SBED) and cosine-based selection (SBCOS) are summarized below. In case of using the SBED method the meta-classifier obtained 92.08% classification accuracy. In case of using the SBCOS method the classification accuracy reached only 89.74%.

We had performed the same experiments after introducing into the meta-classifier the Bayes classifier. The results of the classification accuracy for the meta-classifier containing now 9 classifiers dropped for the SBED method to 90.38% and rose for the SBCOS method to 93.10%.

Another improvement brought to this meta-classifier was in the way the selected classifier chooses the winner class. So we observed that in some cases when the chosen classifier didn't classify correctly the given document, the class from the second position would have been the correct one. So we changed the method for the class selection in the case when all classifiers were rejected because they misclassified similar documents to the given one and forced the classifier to choose the class from the second position if it was very close to the class from the first position. This change improved the classification accuracy of the meta-classifier as follows: in case of using the SBED method the classification accuracy was 93.32% and in case of using the SBCOS method, the classification accuracy was 93.87%.

Hybrid meta-classifier

The idea was to build a meta-classifier which uses 2 components: a non-adaptive considered as a pre-classification stage, and a new adaptive component based on a feed-forward neural network type, regarded as post-classification stage [8].

For our purposes we have implemented a neural network where we can choose the number of neurons contained in the hidden layer and choose the rate of learning. We have used a feed-forward network with back-propagation learning algorithm containing two levels of neurons units with sigmoidal activation function. As input we have a vector with 16 scalars resulting from summing all 9 output vectors from the classifiers and as output we have a vector with 16 binary values (1 for the winner class).

We have performed experiments using different sets for training and testing. We have used decreasing values for the learning coefficient, stopping at certain stages, reducing the rate of learning and then continue the learning. Only when the learning coefficient was reduced we have obtained a small value for the total error of training (average 0,017 per training example). Best results (99.40% classification accuracy) were obtained using a neural network with 192 neurons on the hidden layer. These experiments have proved that the inclusion of a neural network in the meta-classifier makes it more adaptable to the documents that need to be classified, managing to classify documents that other meta-classifiers failed to classify correctly. This new meta-classifier managed to exceed the maximum "theoretical" reachable limit of 98.63% because the neural network has learned even those examples which couldn’t be correctly classified by any of the 9 classifiers.

Main research approaches for clustering of text documents

Our focus in this approach for clustering text documents was the comparison of representation models for text documents and their efficiency in clustering algorithms. We want to bring some “syntax of document” in the document representation which the classical vector space model (VSM) didn’t has.

In our research we have analyzed the opportunity of using the Suffix Tree Document Model (STDM) [5] for representation of text documents. This type of representation is frequently used with the Suffix Tree Clustering Algorithm [2, 9]. We compared two representation models using two different well known clustering algorithms: the hierarchical clustering algorithm (HAC) [4] and the k-Medoids algorithm [3]. We have chosen these two algorithms because they are from two different categories (hierarchical and partitional category) and both use a distance matrix.

In order to reduce the suffix tree size in our approach we build for the STDM representation the suffix tree for any two documents from the document set and then we have computed the similarity / dissimilarity between those two documents. This method has the advantage that the resulting tree size is much smaller than the tree for the entire data set. However this method has the disadvantage to build $n(n-1)/2$ smaller trees, but the construction time required for such a smaller tree is considerably reduced because the trees have few branches and then the search is much faster. Also we need to build and search only in one small tree at a time and therefore the required memory is much smaller, too.

For this representation model we have proposed a new similarity metric called NEWST.

For computing the dissimilarity between documents with the VSM model we have used three known metrics such as: Euclidean distance, Canberra distance and Jaccard distance.

After performing over 150 experiments in which we have compared the results obtained with both representation models and both clustering algorithms we can say that the use of the STDM representation model improves the clustering results.

Our proposed formula for computing the NEWST distance has the following advantages. First: If two documents have no common nodes then the distance between them depends on the number of words that are used to represent those two documents. If the documents are larger and do not even have common nodes, the distance between them will be closer to 1 but still different depending of the documents dimension. Compared with other metrics that always return the value 1 if the documents have no common nodes, this small difference helps us to determine the order in which documents will be merged based on the distance matrix. Thus the large documents will be merged last. The second advantage: If the documents have common nodes we have weighted the returned value with the number of words that are in the common nodes. So we can make the difference between documents that have small common parts and documents that have large common parts. This two advantages offer us the possibility to develop a more accurate clustering.

For example our new developed metric NEWST applied to the STDM model representation for HAC clustering algorithm, obtained for our datasets an improving in clustering accuracy of 34.84% compared to the Jaccard metric used with the VSM representation. The average accuracy for the NEWST metric with the STDM model was 87.23%, compared with 52.39% obtained by the Jaccard metric using the VSM representation model. Also we have performed experiments for testing the influence of applying a stemming algorithm for the STDM model. In these experiments it was observed that for the HAC algorithm, applying the algorithm to extract the roots of words did not modify the clustering results for the STDM representation model.

To verify the proposed metric NEWST we have repeated the same tests with the k-Medoids algorithm. Our metric NEWST obtained a 5.04% improvement compared to the best results obtained by a metric (Jaccard metric) applied to the VSM model. The average accuracy for NEWST using the k-Medoids was 84.20% and for the Jaccard metric with VSM model was 79.16%. Using the stemming algorithm led to an improvement of the results for all metrics used.

Conclusions and further developments

In our paper we have presented some of our research results focused on classification and clustering for text documents. The solutions we have proposed were beneficial for improving the classification and clustering results.

As further work it will be interesting to use several classifiers combining the non-adaptive methods with the adaptive ones, to improve the results without significantly increasing the working time.

In future experiments we will seek to combine classification methods with clustering methods in order to use labeled and unlabeled documents in a hybrid classification algorithm. The idea is to use a small set of labeled data, to guide the clustering algorithm which is trained using a large set of unlabeled data.

Another idea is to change the representation of STDM for clustering algorithms, in order to represent some semantic information contained in the text, which currently is only "present" by keeping order of words in the sentence (phrase).

A major problem that occurs in all clustering and classification algorithms is that they become difficult to use in real situations. For example, clustering algorithms tend to form large clusters over others, which may contain very few documents. This problem occurs because the documents in the same class have some common words and then many documents are grouped together, because each contains some common words that appear in that category.

The classical approach does not consider synonyms as common words. Purely computational approaches to this problem will not lead to dramatic improvements. As a further development, to test the opportunity of representing the meanings of words in documents we intend to use disambiguation algorithms (Word Sense Disambiguation), possibly with the disambiguation algorithms and WordNet meanings of words, and introducing them into the STDM model.

Also we intend to realize a parallelization of the proposed hybrid meta-classifier so that the computing time significantly reduces (in the ACAPS research laboratory at the ULB Sibiu we have the High Performance Computing facilities that enable the successful implementation of these ideas – see: http://acaps.ulbsibiu.ro/index.php/en/).

Acknowledgments

This work was partially supported by CNCSIS-UEFISCSU, project number PN II-RU code PD_670/2010.

References

[1] Crețulescu, R., Morariu, D, Vințan, L, Coman, I., An Adaptive Metaclassifier for Text document, 16th International Conference on Information Systems Analysis, pp. 372-377, ISBN-13: 978-1-934272-86-2(Collection), ISBN-13: 978-1-934272-88-6(Volume II), Florida, USA, 2010.

[2] Janruang, J. Guha, S., Semantic Suffix Tree Clustering, In Proceedings of 2011 International Conference on Data Engineering and Internet Technology (DEIT 2011), Bali, Indonesia, 2011.

[3] Kaufman, L. and Rousseeuw, P.J. Finding Groups in Data: An Introduction to Cluster Analysis, Wiley-Interscience, New York (Series in Applied Probability and Statistics), 1990.

[4] Manning, C., An Introduction to Information Retrieval, Cambridge University Press, 2009.

[5] Meyer,S., Stein, B., Potthast, M., The Suffix Tree Document Model Revisited, Proceedings of the I-KNOW 05, 5-th International Conference on Knowlegdge Management, Journal of Universal Computer Science, pp.596-603, Graz, 2005.

[6] Mitchell, T. Machine Learning, McGraw Hill Publishers, 1997.

[7] Morariu, D., Text Mining Methods based on Support Vector Machine, MatrixRom, Bucharest, 2008.

[8] Morariu, D., Cretulescu, R., Vințan, L., Improving a SVM Metaclassifier for Text Documents by using Naive Bayes, International Journal of Computers, Communications & Control, Vol. V, No. 3, pp. 351-361, ISSN 1841-9836, E-ISSN 1841-9844, September 2010.

[9] Zamir, O, Etzoni, O., Web Document Clustering: A Feasibility Demonstration, Proceedings of the 21st International ACM SIGIR Conference on Research and Development in Information Retrieval, Melbourne, Australia, 1998.