STOPKA: Unbalanced Corpora Classification by Bootstrapping

Yulia Adaskina
InfoQubes
Moscow, Russia
adaskina@gmail.com

Andrey Popov, Polina Rebrova
InfoQubes, Saint Petersburg State University
Moscow, Saint Petersburg, Russia
{hedgeonline, polinka.al}@gmail.com

***Abstract*—The paper describes a tool designed to help the expert to filter out irrelevant documents in cases where the data and classification criteria do not allow any automatic algorithm to be applied. The tool is based on a semi-automatic bootstrapping model that analyses the unlabeled corpus, gets the initial annotation information from the expert and uses it to rank documents according to their similarity to the class in question. Our experiments confirm that the method helps to achieve 0.9 Recall by only viewing around 23% of the corpora.**

I. Introduction

We designed a product which helps to perform a two-class categorization on unbalanced text corpora in Russian. The commercial project we are working on entails a time consuming task of going through a largeunlabelled corpus and annotating a certain amount of texts as irrelevant. The first problem is that there not always explicitly defined criteria for this 'spam' class; the second one is that the class in question constitutes only a small part of the corpus, so there can hardly be an easy automatic way to perform this task. What we were looking for was a tool that would make the labor-intensive and painstaking work of an analyst easier. We designed an application based on the algorithm we developed for the task of semi-automatic extraction of named entities (see [1]). Since we cannot classify the corpus using only statistical methods, we want an algorithm that would choose texts based on their similarity to those marked as target class texts (in our case, spam class texts). Thus, the expert would first annotate a small number of texts, mine relevant features and train the statistical model using those features. The STOPKA application, utilizing LIBSVM Support Vector Machine library, would sort the texts based on their similarity to the class in question and suggest the expert to review them.This way a new portion of target class texts will be obtained and the model can be re-trained with regard to new information.This step can be repeated until some set threshold is reached. Our method was designed to help the expert to filter out irrelevant documents, but we also assessed its work on sentiment classification of tweets.

II. Background

Some major works on text classification using various machine learning techniques are [2] and [3]. The use of bootstrapping for text classification is discussed in[4], [5], [6] and[7],among others.[8] introduces confidence estimate, a way to measure similarity between texts in order to develop expert evaluation of automated methods. Ever since the first introduction of machine learning methods the quest for the optimal feature set continues. Syntactic relations are a relatively new addition to the assortment, and very few researches use it due to the fact that syntactic analysis is costly in terms of effort and performance.[9] is among those who successfully used syntax in their model. We also explored the benefits of syntax for machine learning in [1].

III. Data and Machine Learning Features

The commercial task that triggered this research project concerns the analysis of unsolicited user reviews on major DIY-supermarkets. So, these constitute our primary corpus of 3818 documents. We also tried to assess the applicability of our method to sentiment classification, in order to do so we used the corpus of 8639 tweets on banks and telecommunication companies (the data of SentiRuEval-2015).

Three types of learning features were used, i.e. unigrams, bigrams and syntactic relations. Apart from isolated features, we have also examined the efficiency of feature combinations. Below are all the possible options, with the abbreviations we used for them:

- L – unigrams (lemmas);
- B – bigrams(two adjacent lemmas);
- R – syntactic relations;
- LB – unigrams + bigrams;
- LR – unigrams + syntactic relations;
- BR – bigrams + syntactic relations;
- LBR – unigrams + bigrams + syntactic relations.

Syntactic relations as features are characterized by three elements: two syntactically related lemmas and the relation type between them. We obtain syntactic information using our morphosyntacticparser, which is a part of InfoQubes platform. It parses word sequences and produces syntactic treesfor sentences and distinguishes 20 types of syntactic relations.

As one willsee, different tasksrequire different feature sets to achieve best results, so STOPKA allows the user to tune feature combinations. Binary occurrences is a new feature we are currently working on.

IV. MethodEvaluation

Suchevaluation metrics as Recall and Precision are widely used within natural language processing. Precision may be considered as a characteristic of an automated decision-making

ISBN 978-952-68397-0-7 (paperback), ISBN 978-952-68397-1-4 (PDF)

process, but when we address the task of semi-automatic classification, i.e. the final decision is made by the user, Precision no longer makes sense. Instead, we need somehow to evaluate the degree of operator's involvement in the semi-automatic workflow. In that case, we assume that the more efficient the system is the less human involvement it should require, and so the key metric in our case is Efficiency. For our needs we can define Efficiency as how much Recall we can yield by viewing a fraction of the document set; in other words,Specific Recall: SR = R / F, where R is the Recall, and F is the viewed fraction of the document set. Unfortunately, Specific Recall does not give us the whole picture: we often get the highest Specific Recall after viewing a tiny fraction of our set, which is insufficient for practical purposes. So, we need to compare different SRs according to some predefined Target Recall (i.e. desired Recallto be yielded). In our experiments we set Target Recall as 0.9.

Therefore, the Efficiency in accordance with 0.9 Target Recall is our main parameter to evaluate our method under different settings. We designed an automatic utility to carry out experiments with different settings with gold standard provided requiring no human interaction. We have two different settings; each can affect the performance greatly:

- iteration step (number of documents viewed (and tagged) during each iteration);
- features for our SVM classifier (listed above).

Fig. 1 represents the distribution of Efficiency over Recall for three main feature types: lemmas, bigrams and syntactic features, calculated for telecommunication company category.

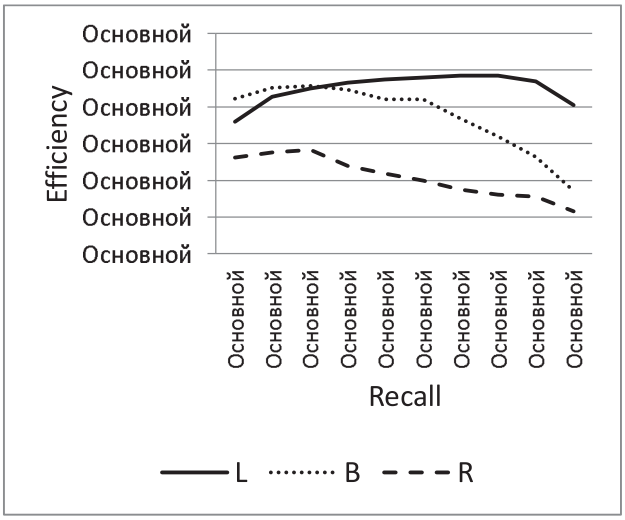

Fig.1. Efficiency distribution over Recall for a telecom brand and different feature sets

Fig. 2 shows what maximum Efficiency can be yielded for different categories by different feature types and their combinations.

V. Results

Our experiments confirmed that the method can be used on our data and it significantly reduces the time the expert needs to spend to filter out irrelevant documents; it can also be used to perform sentiment-based classification. Depending on the task, up to 4.5 Efficiency can be reachedusing STOPKA, which means that 0.9 Target Recall is achieved by only viewing about 23% of the document set. The experiments have also highlighted the importance of syntactic informationfor the overall score. This, however, depends on the text type, as we have shown, the contribution of syntactic relations for the analysis of tweets is insignificant. The explanation is possibly that our syntactic parser works better on texts which are not limited by 140 symbols. One of the main advantages of our method is that machine learning feature setscan be adjusted depending on which Recall level is required, how much time the expert has and what kind of texts is analysed. For the practical use our main evaluation metrics is Efficiency, which can be transformed into Precision for further research. Other topics for further exploration include the use of semantic tags as machine learning features and expansion of our method for more complex classification models.

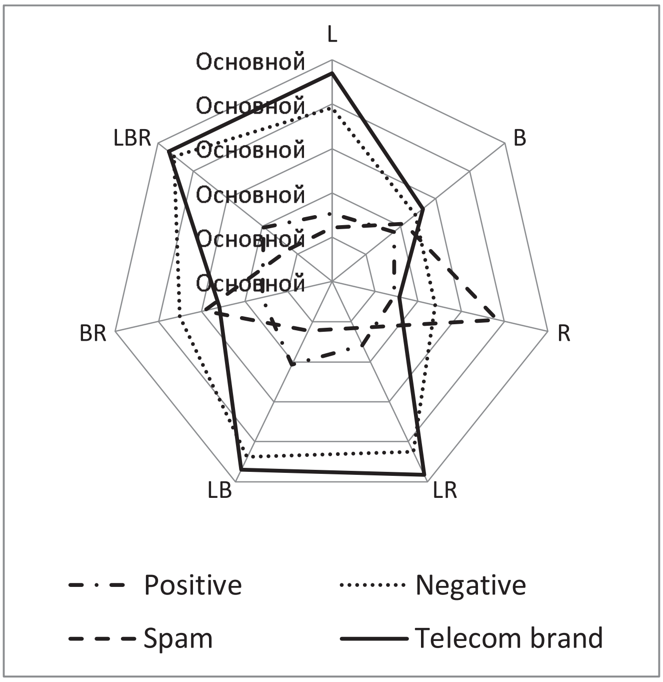

Fig.2. Maximum Efficiency for different feature sets and different categories

References

[1] Adaskina Yu. V., Panicheva P. V., Popov A. M. (2014), Semi-Automatic Lexicon Augmenting Based on Syntactic Relations [Poluavtomaticheskoyepopolneniyeslovareynaosnovesintaksicheskik hsvyazey], *In Proc. of Internet and Modern Society Conference*, Saint Petersburg, pp. 271–276.

[2] Maron M. Automatic indexing: an experimental inquiry // In *J. Assoc. Comput. Mach.* 1961. Vol. 8, №3. P. 404 – 417

[3] Sebastiani F. Machine Learning in Automated Text Categorization // In *ACM Computing Surveys (CSUR)*. 2002. Vol. 34, №1. P. 1 - 47.

[4] McCallum A., Nigam K. Text Classification by Bootstrapping with Keywords, EM and Shrinkage // In *ACL Workshop on Unsupervised Learning in Natural Language Processing*. 1999. P. 52 – 58.

[5] Gliozzo A., Strapparava C., and Dagan I. Improving text categorization bootstrapping via unsupervised learning // In *ACM Transactions on Speech and Language Processing (TSLP)*. 2009. Vol. 1, №1.

[6] Jones R., McCallum A., Nigam K., Riloff E. Bootstrapping for Text Learning Tasks // In *IJCAI-99 Workshop on Text Mining: Foundations, Techniques and Applications.* 1999. P. 52 – 63.

[7] Nigam K. *Using Unlabeled Data to Improve Text Classification*. Doctoral Dissertation, Computer Science Department,Carnegie Mellon University. Technical Report CMU-CS-01-126. 2001.

[8] Berardi G, Esuli A, Sebastiani F. A utility-theoretic ranking method for semi-automated text classification // In*The 35th International ACM SIGIR Conference on Research and Development in Information Retrieval: Proceedings of the Conference.* 2012. P. 961 –970.

[9] Furnkranz, J., Mitchell, T., Riloff E. A Case Studyin Using Linguistic Phrases for Text Categorization on the WWW, AAAI/ICML // *Workshop on Learning for Text Categorization*. 1998.