PROOF OF CONCEPT STATISTICAL SENTIMENT CLASSIFICATION AT ROMIP 2011

Poroshin V. (vladimir.poroshin@m-brain.com)

M-Brain Oy, Helsinki, Finland

In this paper we present a simple statistical classification method that predicts whether the opinion expressed by text in natural language is positive or negative. There are two main approaches in the sentiment or opinion detection: linguistic rule based systems and statistical algorithms. While statistical methods are easier to build when sufficient training data is available, it is widely perceived that a linguistic system can deliver better results. Our work was intended to prove the concept that a simple Naïve Bayes based statistical classification algorithm with a minor language dependent adaptation is able to perform well in a binary sentiment classification task. In order to prove the hypothesis, we participated in Russian Information Retrieval Seminar (ROMIP) 2011 sentiment classification track [1], and achieved quite competitive results in sentiment prediction of Russian blog posts. This paper contains a detailed description of our classification method, including a feature extraction and normalization process, training and test data, evaluation metrics; and presents our official ROMIP results.

Keywords: statistical sentiment classification, sentiment analysis, sentiment detection, opinion mining, Naïve Bayes, ROMIP

1. Introduction

The goal of this work is to compare a simple statistical based approach for a sentiment classification problem to other statistical and linguistic methods in the scope of the ROMIP 2011 sentiment classification track [1]. The task of the sentiment analysis in our context lies in automatic categorization of incoming text in Russian into two classes: positive or negative in general, i. e. without any specified target of the sentiment. This is one variation of the sentiment classification problems, which in general can vary in number of classes to predict (2, 3, 5 or more classes, including neutral and other emotional states such as angriness, sadness and so on) or in the target of the sentiment (specific word, sentence, whole text, etc.). Although, a binary 'no target' sentiment classification in many cases is simpler than the other sentiment classification tasks; it can serve as a basis for implementation of some of them.

Sentiment analysis can be viewed as a classification problem where one can use well-known statistical classifiers, as it is outlined in a number of publications [2, 4]. Our research aims to show that a simple statistical classifier with a pretty generic feature extraction process can achieve good results in the sentiment classification.

The paper is organized in the following way: section 2 contains the review of a modified Naïve Bayes classifier; section 3 describes features and their extraction process; section 4 illustrates test and training data; section 5 presents our official ROMIP 2011 evaluation results; and, finally, section 6 completes the paper with conclusions.

2. Method description

In order to assign sentiment labels to new test documents we use Naïve Bayes algorithm with few modifications as our classification method.

In multinomial Naïve Bayes [5] a class C is assigned to a test document d, where

$$C = argmax_c P_{NB}(c|d)$$

$$P_{NB}(c \mid d) = \frac{P(c)\sum_{i=1}^{m} P(f \mid c)^{n_i(d)}}{P(d)} \quad (1)$$

where f is one of m features, $n_i(d)$is the count of feature f_i in a document d. $P(c)$ and $P(f|c)$ are calculated through maximum likelihood estimates, and an add-1 smoothing is utilized for unseen features.

With the focus on the improvement of a standard multinomial Naïve Bayes we applied the following modifications: a term frequency (TF) transformation (1.1) and a TF transformation based on length (1.2).

In a TF transformation all term frequencies $n_i(d)$ in the formula (1) are replaced by:

$$n_i(d) = \log(n_i(d)+1) \quad (1.1)$$

It has been shown in [3] that this transformation models the term distribution of the text in a better way by reducing the weight of frequent features.

Other TF transformation normalizes feature counts to deal with the negative effect of long documents, since Naïve Bayes assumes independence of features [3]. For this we transform counts $n_i(d)$ to

$$n_i(d) = \frac{n_i(d)}{\sqrt{\sum_k n_k(d)^2}} \quad (1.2)$$

3. Features

We have tried several types of features, including words uni-grams, bi-grams, word-forms, stems and lemmas. The best found combination was to use uni- and bi-grams of lemmatized words. Stop-words were also removed from the text.

The process of lemmatization lies in the determination of a dictionary form (or lemma) of a given word. In many languages, including English and Russian,

a word-form can have more than one lemma when it is considered independently from the context. For instance, a Russian word form *моя* can be lemmatized to a possessive pronoun *мой* (*mine*) or to a verb *мыть* (*to wash)*. Usually, selection of the correct lemma can be deduced from the context sentence with the help of the part of speech tagger or some other linguistic processing. In our case we use a frequency dictionary to select the most probable lemma. In case lemmas of some word-form are not presented in the dictionary, random ones are picked.

In terms of the nature of the sentiment classification task, we have found that weight of negative particles (such as *not* in English) in our method usually dominates in case they are present. For instance, a phrase "*it is not bad*" will be classified as negative, because words *not* and *bad* have quite high frequencies in the training data with negative sentiment labels. As a result many test documents can be incorrectly classified as negative ones only because they contain a negative particle. To overcome this problem we used a data pre-processing step that glues a negative particle to the word next to it. In our example, it will become "*it is notbad*".

Our full feature extraction process is the following:

- replace all URLs to a special label *tokenurl*
- replace all positive and negative emoticons to special labels *tokensmilepositive* and *tokensmilenegative* correspondingly
- lowercase all text
- remove repeated letters
- remove stop words
- glue negative particles as it was described
- lemmatize each word
- collect uni- and bi-grams as features

"Remove repeated letters" step becomes quite necessary when we deal with the text from social media sources, for example, from Twitter. There the words are usually misspelled by repeating the letters, for example, "*woooo!!!! Suuuch a messsss! brrrrr....*". During this step we delete all letters that appear in a word more than 2 times, i. e. in our example the saying will be transformed to "*woo!!!! Suuch a mess! brr....*".

4. Test and Training data

Our training data was collected from three sources: the web site http://lovehate.ru, Yandex market http://market.yandex.ru and Twitter.

Main portion of the training data was collected from the web site lovehate.ru, which contains opinions in Russian on various topics. There people mark themselves their comments on some topics as positive or negative.

From Twitter we got a sample of positive and negative tweets in Russian by collecting tweets from the Twitter API with emoticons as a query. A tweet with a negative emoticon, such as :(:-(is considered to have a negative sentiment, and a tweet with a positive emoticon — a positive one [4]. We used only a small portion of such data

for training of the classifier because of the low quality of it. The decision to include the Twitter training data was connected with intention to have internet slang words in our model.

The last set of the training data is a dump of positive and negative opinions about digital cameras in corresponding Yandex market web pages [6]. This data was received from ROMIP as a part of the in-domain training set, since one of the ROMIP 2011evaluation topics in the sentiment classification track was about digital camera products.

We did not use other official ROMIP 2011training data.

A summary of the training data sources is presented in the table 1.

Table 1. Training data

	Number of topics	Total number of words	Total number of samples	Is it in-domain?
lovehate.ru	2 850	20 267 645	346 041	No
ROMIP Yandex market digitalcam	1	602 101	19 986	Yes
Twitter	N/A	2 527 064	63 511	No

As participants of the ROMIP 2011 we also received a test data, which includes a set of blog posts on 3 topics: digital cameras', books' and movies' reviews. In the evaluation of our method we considered only digital cameras testing set, because we used a corresponded in-domain training data only for this topic. Test documents were manually judged by two assessors in order to create a human quality sentiment labels for evaluation. For simplicity we used evaluation results calculated only for test samples where both assessors assigned the same labels (table 2).

5. Evaluation results

Official ROMIP evaluation results of our algorithm (denoted as *stats*) are shown in the table 2. Based on average F-Measure score the proposed statistical method is on the 4th place out of 25 total results. Average of all participants' results is also presented in the table 2.

Table 2. Official ROMIP 2011 results for 2 class sentiment classification track for the digital cameras topic (first 5 out of 25 total runs by participants sorted by F-Measure_AND ***F*** and an average over all runs)

	P	R	F	A	P_P	P_N	R_P	P_N	F_P	F_N
xxx-24	0.9092	0.9337	0.9209	0.9569	0.9811	0.8372	0.9674	0.9	0.9742	0.8675
xxx-9	0.8905	0.9291	0.9082	0.9490	0.9810	0.8	0.9581	0.9	0.9694	0.8471
xxx-16	0.9355	0.88052	0.9052	0.9529	0.9593	0.9118	0.9860	0.775	0.9725	0.8378
stats	**0.8562**	**0.8416**	**0.8486**	**0.9216**	**0.9493**	**0.7632**	**0.9581**	**0.7250**	**0.9537**	**0.7436**
xxx-6	0.8059	0.8808	0.8356	0.9020	0.9703	0.6415	0.9116	0.85	0.9400	0.7312
average	0.7467	0.7692	0.72156	0.81522	0.94716	0.54615	0.83134	0.707	0.8741	0.5690

Average precision P, recall R and F-measure F are calculated as:

$$P = \frac{P_N + P_P}{2}, \quad R = \frac{R_N + R_P}{2}, \quad F = \frac{F_N + F_P}{2}$$

Where precision, recall and F-measure for a positive class:

$$P_P = \frac{tp}{tp + fp}, \quad R_P = \frac{tp}{tp + fn}, \quad F_P = 2\,\frac{P_P \cdot R_P}{P_P + R_P}, \quad \text{where}$$

tp — number of true positives, fp — number of false positives, fn — number of false negatives.

And for a negative class:

$$P_N = \frac{tn}{tn + fn}, \quad R_N = \frac{tn}{tn + fp}, \quad F_N = 2\,\frac{P_N \cdot R_N}{P_N + R_N}, \quad \text{where}$$

tn — number of true negative

Total accuracy of the method is:

$$A = \frac{tp + tn}{tp + tn + fp + fn}$$

6. Conclusion

We presented a statistical classifier for a two class sentiment classification task. Our method is based on slightly modified Naïve Bayes classification algorithm and a simple linguistic data pre-processing, which helps to better suit the domain of the problem.

Our participation in ROMIP 2011 sentiment classification track serves as the evaluation of the proposed method. The results show that our method is competitive enough in comparison to other participants' approaches. This means that even a quite simple statistical method can show good performance in this type of tasks.

In the future perspective, our algorithm can be extended and further improved in several different ways. Some of the domain adaptation techniques such as mixture of models can be used to achieve better results for the data with a predefined topic.

Feature extraction process could be also improved by employing a proper stemming technique, which is able to resolve words' ambiguity. Also, other classification models like SVN may perform better in this problem because they don't assume independence of features and better model the data.

The idea to glue negative particles that was described in section 3 also needs an improvement. Not an every word next to negative particle is a target of gluing. There could be words in between, for example, "it is *not* so *bad*". Employing word dependencies from a syntactical parser will help to find correct targets.

Solution to the two class sentiment problem (positive/negative) can be as well further embedded in a framework, where a neutral class is detected also, for example, with the help of another classifier, which detects neutrality of the text.

References

1. *ROMIP*: Russian Information Retrieval Evaluation Seminar, http://romip.ru
2. *B. Pang, L. Lee, and S. Vaithyanathan.* Thumbs up? Sentiment classication using machine learning techniques. [Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP) 2002], pp. 79–86
3. *Jason D. Rennie, Lawrence Shih, Jaime Teevan, David R. Karger.* Tackling the Poor Assumptions of Naive Bayes Text Classifiers. [Proc. of ICML'2003]. pp.616~623
4. *Go, A., R. Bhayani, and L. Huang.* Twitter sentiment classification using distant supervision. Technical report, Stanford Digital Library Technologies Project 2009
5. *C. D. Manning and H. Schutze* (1999). Foundations of statistical natural language processing. MIT Press
6. *Yandex* Market: http://market.yandex.ru/