A Comparison of Machine Learning Algorithms in Opinion Polarity Classification of Customer Reviews

KRUNOSLAV ZUBRINIC
University of Dubrovnik
Department of Electrical engineering and computing
Cira Carica 4, Dubrovnik, CROATIA
krunoslav.zubrinic@unidu.hr

MARIO MILICEVIC
University of Dubrovnik
Department of Electrical engineering and computing
Cira Carica 4, Dubrovnik, CROATIA
mario.milicevic@unidu.hr

TOMO SJEKAVICA
University of Dubrovnik
Department of Electrical engineering and computing
Cira Carica 4, Dubrovnik, CROATIA
tomo.sjekavica@unidu.hr

INES OBRADOVIC
University of Dubrovnik
Department of Electrical engineering and computing
Cira Carica 4, Dubrovnik, CROATIA
ines.obradovic@unidu.hr

Abstract: In this paper we analyze reviews written by customers of an online shop, by employing opinion polarity classification on document level using five machine learning algorithms: Naïve Bayes, Support Vector Machine, Neural networks, C4.5 algorithm and classifier based on maximum entropy. We achieved the best results using Support Vector Machine algorithm (accuracy=0.845) and maximum entropy classifier (accuracy=0.84). Although those results are not as good as results that can be achieved in topic-based categorization, compared to similar researches in opinion polarity classification, they indicate a relatively good predictive performance of classical machine learning algorithms.

Key–Words: opinion polarity classification, sentiment analysis, natural language processing

1 Introduction

In today's digital economy, brand and reputation have become interlinked, because of the growing power of social networks and networking websites. Companies that want to succeed in such an environment invest considerable resources in digital reputation marketing.

Understanding customer sentiments on a product and their readiness to recommend that product to others became extremely important. Those data can give insight on how customers perceive specific products and provide usable information for products improvements. Companies often use customer reviews on online review sites and social networks as a source of such information [1].

That kind of information can be categorized into two basic types: facts and opinions. Facts represent objective expressions of customers about different events, entities or their properties, while opinions are usually subjective expressions that represent sentiments or emotions which customers have towards the product [2].

Most of such content is in textual form, and manual tracking and evaluation of that information on many websites can be very demanding. An information system that can process such information can be very helpful.

Sentiment analysis of text is a field in Natural Language Processing (NLP) that analyzes different subjective information such as opinions, sentiments, and emotions based on observations of peoples actions captured using their writings [3].

In the last 20 years, with the rise of machine learning methods in NLP and Information retrieval (IR), this area has begun to develop considerably. As more data becomes available, more ambitious problems are actualized. The development of the systems that can effectively process subjective information has become an important research goal.

In traditional text categorization, a goal is to classify documents by topic in a set of categories given by definition, or specified by a researcher. There can be many categories, and they are mostly application-dependent and diverse in different domains.

One of the most popular tasks in sentiment

analysis is opinion polarity classification where the main goal is to position an opinionated text on the scale of polarities [4].

Compared to traditional categorization, opinion polarity classification is usually binary, and it labels opinions as ”positive” or ”negative”. Such categorization sometimes uses the third class for ”neutral” opinion. The meaning of that class is not unambiguous, as they sometimes use it as a label for ”objective” text, but in other cases, it can be a strong opinion that something is ”mediocre” [4].

Opinion polarity classes are the same, or very similar in different domains and, for example, we use same classes when expressing our opinions on the quality of movies and quality of coffee we drank.

In terms of granularity, a subject of research could be the polarity of a document as a whole, each sentence in a document, or a specific aspect of a product specified in a document [3].

One document can contain paragraphs or sentences with different, sometimes opposing sentiments. In such cases, the overall sentiment of the document is a function of the set of sentiments at the sub-document level [4].

In this paper, we will evaluate the results achieved using five classical data mining algorithms for supervised learning on a task of opinion polarity classification of reviews written by customers of an online shop.

This paper is organized as follows. After the introduction, the second section gives a brief overview of related work in the area of opinion polarity classification. The methodology used in our experiments is described in the third section. The achieved results are presented and interpreted in the fourth section. Finally, our conclusions and a description of possible follow-up studies are described in the fifth section.

2 Related Work

Supervised classification methods apply machine-learning algorithms on a set of training data to predict the label of unseen test data. For high-quality results, they require a large amount of annotated data for training [3].

Opinion polarity classification depends on annotated opinionated data sets. Web 2.0 technologies, such as social media and networking sites, has played an important role in providing researchers with a large amount of opinionated user-generated content.

An important step in the classification process is extracting a set of right features. In NLP, the basic features are words or groups of words. However, the task of choosing right features from a text is harder than it looks. In an experiment, Pang et al. [5] asked two human subjects to pick keywords from opinionated texts that, in their opinion, could be good indicators of positive or negative sentiment. The percentage of correctly classified documents using keywords chosen that way was 58–64%. Researches later confirmed this finding.

Many studies in this field look at document-level sentiment analysis as a special case of text categorization task. This approach represents a text with a feature vector where features are individual terms. Those studies use standard machine learning methods, usually supplemented with NLP processing and sentiment specific features [3].

Other researchers use lexicons that associate words with sentiment categories or describe a structure of a document using an analysis of sub-document units. Recent approaches use neural networks and deep learning [5].

Sebastiani [6] described a general methodology for automated classification of texts and compared the characteristics of basic categorization algorithms in text mining. Pang et al. [5] compared performances of Support Vector Machine (SVM), Naïve Bayes (NB) and maximum entropy classifier on the sentiment classification of movie reviews.

Sentiment analysis is a complex problem and many text specific features, such as domain, term presence, n-grams, part of speech (POS), syntax, negation, and topic-orientation have a significant impact on the results [3], [4].

Several studies have shown that the domain and a context of document can have an influence on the accuracy of sentiment classification, as the same phrase can indicate different sentiments in different domains [4] or contexts [7]. To minimize this impact some researchers follow a simple approach using only features that are good subjectivity indicators in all observed domains [8], while others use complex models that choose features on the base of correlations between the pivot features found in all domains and all other features [9].

The important difference between the sentiment analysis and the classic information retrieval is that, in sentiment analysis, the presence of a term has a significantly greater influence than its frequency [5].

Usage of n-grams as features is still a matter of debate because some researches report that unigrams outperform bigrams in sentiment classification [5], while others find that in some settings the use of bigrams and trigrams gives better results [10].

POS information is commonly exploited in sentiment analysis as some types of words (such as

adjectives, nouns, and verbs) are more likely to carry information about the sentiments expressed in a text [3].

Using negating words may flip the polarity of a sentence, and it is important to identify such cases. A simple approach to identifying a change of polarities is attaching the word "NOT" to words occurring close to negation terms [11]. Other researchers try to optimize this technique using the POS tags to mark the complete phrase as a negation phrase [12].

The additional problem arises from specific subtle expressions in the human languages, such as sarcasm or irony [4].

3 Methodology

In this study, we formulated this problem as binary categorization of documents that express positive or negative opinions about the recommendation of a particular product. We have been categorizing documents from the domain of product reviews, at the document level. Earlier studies described in the previous section, inspired this approach.

As a dataset, we used the set of reviews written by customers of women clothing in an online shop [13]. According to the author of the dataset, set contains real, anonymized commercial data.

This dataset had 23,486 rows and 10 feature variables in one CSV file. Each row corresponds to one customer review, and contains the following variables: "clothing ID", "reviewers age", "title of the review", "review text", "rating", "recommendation ID", "positive feedback count", "product division", "product department" and "product class name".

We classify customer recommendations of a product in two classes — "recommended" and "not recommended" — based only on a title and a text of a review.

The variable "recommendation ID" is used as a label of classes, where the value "1" means a positive recommendation, and value "0" means a negative recommendation.

During the preprocessing of the dataset, using a simple Python script we removed all variables except "title of the review", "review text" and "recommendation ID", and deleted 844 rows where attributes "title of the review" and "review text" were empty.

The frequency distributions of classes in recommendation are imbalanced, as 81.89% of documents are labeled as "recommended". Such distribution could be a problem as the model may grow a biased classification towards "recommended" classes. In order to equalize the number of examples for learning classifiers, we randomly choose 8,000 reviews — one half from subset labeled as "recommended", and another half from subset labeled as "not recommended".

We stored each review as a separate document, depending on their label, in folders titled "recommended" or "not-recommended".

For data analyses, we used the WEKA data mining and machine learning tool [14], along with five WEKA's default implementations of algorithms which have shown good results in opinion polarity classification: Naïve Bayes [15], Sequential minimal optimization algorithm (SMO) [16], Multilayer perceptron neural network (MLP) [17], J48 implementation of C4.5 decision tree [18] and Logistic, a classifier based on a maximum entropy [19].

In the preparation phase, we tokenized each document on spaces and punctuation marks, removed all remaining special characters, and changed all letters to lowercase.

At the end of this phase, each document was represented in the form of a bag of features.

As the starting number of features was large, before the classification phase we filtered attributes using a feature selection algorithm [20] based on the correlation of the presence of features in the set of documents.

We conducted two experiments. In the first experiment, we used single words as features. We removed stop-words from documents using a list of commonly used stop words in the English language [21], and stem words using Lovins stemmer [22]. After filtering, the number of attributes was reduced to 46.

In the second experiment, we used n-grams as features. Each n-gram contained 1 to 3 words. Stop-words were not removed from documents, nor were words stemmed. The filtering process reduced the number of attributes to 76.

Validation of models was performed using 10-fold cross-validation on the same dataset as learning.

Class distribution in the dataset is balanced and we measured the performances of each classifier using precision (P), recall (R), F_1, and accuracy [23].

4 Results and Discussion

Table 1 shows the results of our experiments.

In the first experiment where we preprocessed data using basic NLP methods, Logistic, a classifier based on a maximum entropy had the highest accuracy

Table 1: Results of the classification

Algorithm	Accuracy	
	1st experiment	2nd experiment
Naïve Bayes	0.804	0.825
SMO	0.807	**0.845**
MLP	0.785	0.813
J48	0.794	0.805
Logistic	**0.808**	0.840

of 0.808. The SMO algorithm based on SVM achieved only a slightly lower result of 0.807.

The MLP algorithm based on the neural networks achieved the worst results, and one reason could be the small dataset used for learning in this experiment.

In the second experiment, we included bigrams and trigrams in the set of features, and classifiers worked with slightly more features than in the first experiment. Overall the results are better than the results of the first experiment, and two classifiers that achieved the best results are the same (only in a different order). The best results were achieved by the SMO algorithm with an accuracy of 0.845, and the Logistic classifier with an accuracy of 0.840.

In this experiment, the worst results were achieved by the J48 algorithm that generates a decision tree. Such a result is not unexpected because results reported in the earlier studies indicated that decision trees in some cases perform rather bad in text classification [6].

Figures 1 and 2 show the results of the experiments, each of the classes is represented by values of precision, recall and F_1 indicators.

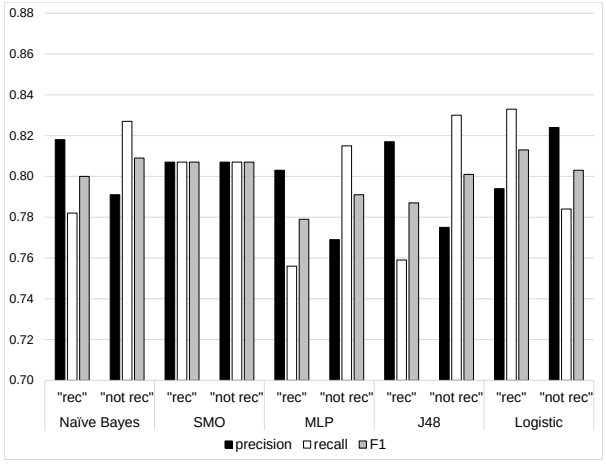

Figure 1: Results of the first experiment

The results of the first experiment are fairly uniform. The SMO algorithm achieved the best results, as it produced identical results in both classes ($P=R=F_1=0.807$).

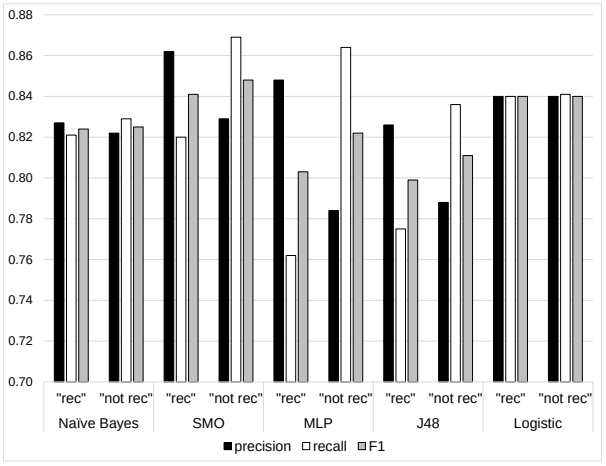

Figure 2: Results of the second experiment

If we observe the results of the second experiment, we can notice the same tendency of results inside classes as in the previous experiment. Precision and recall are harmonized, and the most balanced results between classes were produced by the Logistic classifier ($P=R=F_1\approx0.84$).

On the base of achieved results, we can conclude that classifiers were not biased towards any class, and that is partly the result of a balanced choice of data.

The results achieved in the second experiment are similar to results described in other researches in opinion polarity classification using data mining techniques with the minimal use of NLP techniques. They show a relatively high predictive performance of simple data mining classifiers in the opinion polarity classification tasks.

5 Conclusion

In this research, we demonstrated opinion polarity classification using five standard data mining algorithms. The achieved results indicate a relatively high-performing predictive performance of simple classifiers that can be used in opinion polarity classification without complex features engineering.

Since the experiments were conducted on a relatively small dataset, in future researches they could be repeated on a larger set of data, while model testing could be done on a dataset from another source in the same domain, or on a dataset from another domain.

In addition, it would be interesting to see how this model behaves in the classification of texts written in languages other than English.

References:

[1] D. Ryan, *Understanding Digital Marketing: Marketing Strategies for Engaging the Digital Generation*, 4th ed., Kogan Page, 2017.

[2] B. Liu, Sentiment Analysis and Subjectivity, in *Handbook of Natural Language Processing*, 2nd ed., CRC Press, 2010, pp. 627–666.

[3] A. Yadollahi, A. G. Shahraki, and O. R. Zaiane, Current State of Text Sentiment Analysis from Opinion to Emotion Mining,*ACM Computing Surveys*, vol. 50(2), 2017, pp. 25–33.

[4] B. Pang and L. Lee, Opinion Mining and Sentiment Analysis,*Foundations and Trends in Information Retrieval*, vol. 2(1-2), 2008, pp. 1–135.

[5] B. Pang, L. Lee, and S. Vaithyanathan, Thumbs Up?: Sentiment Classification Using Machine Learning Techniques, in *Proceedings of the ACL Conference on Empirical Methods in NLP (EMNLP)*, 2002, pp. 79–86.

[6] F. Sebastiani, Machine Learning in Automated Text Categorization, *ACM Computing Surveys*, vol. 34(1), 2002, pp. 1–47.

[7] T. Mullen and N. Collier, Sentiment Analysis Using Support Vector Machines with Diverse Information Sources, in *Proceedings of the EMNLP*, 2004, pp. 412–418.

[8] H. Yang, L. Si, and J. Callan, Knowledge Transfer and Opinion Detection in the TREC2006 Blog Track, in *Proceedings of the 15th Text REtrieval Conference(TREC)*, 2006.

[9] J. Blitzer, M. Dredze, and F. Pereira, Biographies, Bollywood, Boom-boxes and Blenders: Domain Adaptation for Sentiment Classification, in *Proceedings of the 45th Annual Meeting of the ACL*, 2007, pp. 440–447.

[10] K. Dave, S. Lawrence, and D. M. Pennock, Mining the Peanut Gallery: Opinion Extraction and Semantic Classification of Product Reviews, in *Proceedings of the 12th International Conference on WWW*, 2003, pp. 519–528.

[11] S. Das and M. Chen, Yahoo! for Amazon: Extracting Market Sentiment From Stock Message Boards, in *Proceedings of the APFA Annual Conference*, 2001.

[12] J-C. Na, H. Sui, C. Khoo, S. Chan, and Y. Zhou, 2004. Effectiveness of Simple Linguistic Processing in Automatic Sentiment Classification of Product Reviews, in *Proceedings of the 8th International ISKO Conference*, 2004, pp. 49–54.

[13] N. Brooks, Women's E-Commerce Clothing Reviews Dataset, ver. 1, January 2018, [Online]. Available: `https://www.kaggle.com/nicapotato/womens-ecommerce-clothing-reviews`. [Accessed 15th June, 2018]

[14] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, The WEKA Data Mining Software: An Update, in *SIGKDD Explorations*, vol. 11(1), 2009, pp. 10–18.

[15] G. H. John, P. Langley, Estimating Continuous Distributions in Bayesian Classifiers, in *Proceedings of the 11th Conference on Uncertainty in AI*, 1995, pp. 338–345.

[16] S. S. Keerthi, S. K. Shevade, C. Bhattacharyya, K. R. K. Murthy, Improvements to Platt's SMO Algorithm for SVM Classifier Design, *Neural Computation* vol. 13(3), 2001, pp. 637–649.

[17] L. Atlas et al., A Performance Comparison of Trained Multilayer Perceptrons and Trained Classification Trees, in *Proceedings of the IEEE*, vol. 78(10), 1990, pp. 1614–1619.

[18] J. R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann, 1992.

[19] S. le Cessie, J. C. van Houwelingen, Ridge Estimators in Logistic Regression, *Applied Statistics* vol. 41(1), 1992, pp. 191-201.

[20] M. A. Hall, Correlation-based Feature Subset Selection for Machine Learning, Ph.D. dissertation, The University of Waikato, Hamilton, New Zealand, 1999.

[21] Stopwords ISO, [Online]. Available: `https://github.com/stopwords-iso/stopwords-en`. [Accessed 21th June, 2018]

[22] J. B. Lovins, Development of a stemming algorithm, *Mechanical Translation and Computational Linguistics* vol. 11, 1968, pp. 22–31.

[23] C. J. van Rijsbergen, *Information Retrieval* 2nd ed., Butterworth-Heinemann, 1979.