

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/318056374>

Comparison of Naive Bayes, Random Forest, Decision Tree, Support Vector Machines, and Logistic Regression Classifiers for Text Reviews Classification

Article · January 2017

DOI: 10.22364/bjmc.2017.5.2.05

CITATIONS

257

READS

28,715

2 authors:



Tomas Pranckevicius
Vilnius University

2 PUBLICATIONS 292 CITATIONS

[SEE PROFILE](#)



Virginijus Marcinkevičius
Vilnius University

58 PUBLICATIONS 655 CITATIONS

[SEE PROFILE](#)

Comparison of Naïve Bayes, Random Forest, Decision Tree, Support Vector Machines, and Logistic Regression Classifiers for Text Reviews Classification

Tomas PRANCKEVIČIUS, Virginijus MARCINKEVIČIUS

Vilnius University, Institute of Mathematics and Informatics
Akademijos str. 4, Vilnius, Lithuania

`{tomas.pranckevicius, virginijus.marcinkevicius}@mii.vu.lt`

Abstract. Today, a largely scalable computing environment provides a possibility of carrying out various data-intensive natural language processing and machine-learning tasks. One of these is text classification with some issues recently investigated by many data scientists. The authors of this paper investigate Naïve Bayes, Random Forest, Decision Tree, Support Vector Machines, and Logistic Regression classifiers implemented in Apache Spark, i.e. the in-memory intensive computing platform. The focus of the paper is on comparing these classifiers by evaluating the classification accuracy, based on the size of training data sets, and the number of n -grams. In experiments, short texts for product-review data from Amazon¹ were analyzed.

Keywords: Machine Learning, Naïve Bayes, Random Forest, Decision Tree, Support Vector Machines, Logistic Regression, Apache Spark, Natural Language Processing

1 Introduction

Data classification is an area investigated by many data scientists, with the demand for a data classification set to continue growing in the future for a number of reasons: firstly, for detecting antisocial online behavior, antisocial users in a community, or that which act strangely or even appear dangerous (Cheng et al., 2014); secondly, classification allows the investigation of global social and information networks to gather special knowledge derived from hundreds millions of users around the globe; thirdly, for analyzing media generated in social communities, including images, videos, sound and text, and to group users in relation to their locations, networks of friends, hobbies, activities, and professions. The main goal of text classification is to identify and assign the predefined class to a selected instance, when the training set of instances with class labels is given. Classification methods are unique data-processing features of machine learning (Alpaydin, 2010) and allows to run multi-class text-classification. Text classification into predefined classes can be recognized as sentiment or polarity analysis that indicates the emotional tone for a given content and assigns the meaning of

¹ Amazon is registered trademark. More: <https://amazon.com>

sentiment e.g. either positive or negative. Application of sentiment analysis can be used almost in every aspect of the modern world from products and services such as healthcare, online retail, social networks, to financial services or political elections, and other possible domains where humans leave their feedback. Organizations usually are seeking to collect consumer or public opinions about their products and services. For that, many surveys or opinion gathering techniques and methods are conducted with the focus to targeted groups or by using any other information that is available. Therefore, developed concepts and techniques of informatics engineering can suggest modern solutions including sentiment analysis that explores topics such as classification with machine learning and works with collections of humans' opinions or customer feedback data expressed within short text messages, e.g. product-reviews.

The results of this investigation can be used in a variety of large scale textual data processing systems and tools, finding the optimal structures and their values to implement the algorithms, understand and predict the data to support decision making and knowledge gathering process, i.e. to classify unclassified product-review data that will help the customer to decide whether to order products and services or not.

With the intention to process text classification, firstly text corpus preparation must be considered by using special natural language processing features, such as: 1) *bags of words* in combination of *n*-grams (Zhang et al., 2010); 2) *segmentation* by separating each single word with punctuation or white space (Grefenstette and Tapanainen, 1994), removing all stop words, such as *a* and *the*, or by making all capital letters a lower case (Daudaravičius, 2012); 3) *stemming* by reducing words to their stemma forms (Frakes et al., 1992); 4) *term frequency* by counting the frequency of words which helps to identify how important a word is to a document in a corpus 5) *word embedding* is transformation of words to an array of numeric values of semantic or contextual information that computer can understand.

In our research, big data-classification tasks will be completed by using the MLlib library on the Apache Spark computing platform. Apache Spark is an in-memory computing platform designed to be one of the fastest computing frameworks able to run various kinds of computing tasks. The Apache Spark project was started on May 30, 2014. The platform is an extension of Hadoop MapReduce (Gu et al., 2013) that supports interactive queries and stream processing. In contrast to Hadoop MapReduce, Apache Spark can run all computations in memory rather than only on disc (Karau et al., 2015). Such intensive in-memory computations open the door to classification methods that are effective in solving big-data multi-class text-classification tasks.

In this paper, Naïve Bayes (Manning et al., 2008), Random Forest (Agrawal et al., 2013), Decision Tree (Rokach et al., 2005), Support Vector Machines (Flannery et al., 2007), and Logistic Regression (Caraciolo, 2011) classifiers are used to solve multi-class classification tasks. So that to investigate these methods and identify the optimal number of *n*-grams (Cavnar et al., 1994), and to get the best classification accuracy (Ivanov, 1972) using product-review data taken from Amazon. These methods are the most popular and accurate multi-class classification methods in the given research domain. Deep learning methods such as deep neural networks have much bigger algorithm capacities, thus we consider comparing methods that has similar algorithm capacity. Following that, artificial neural networks can train themselves and define the multi-layer

relationships between features of the objects. In opposite to the classical classification methods, features are constructed by human intervention as part of separate process. Therefore, feature selection, and classification are as component parts of classical classification methods.

This paper is organized as follows: section 1 presents an introduction to machine-learning technologies and classification methods that are used for text classification, section 2 describes the workflow model and feature selections, section 3 illustrates the results of experiments, and section 4 presents conclusions.

2 Workflow model and feature for reviews processing

Amazon customers' product-review data for Android Apps is selected for investigating (McAuley et al., 2015). The total number of records is given by $n = 2638274$. The customers' review fields were extracted: *review text* – a written customer review about the product; *overall* – a rating given by the customer for the product (*ratings from 1 to 5 are used in this research: 1 is the lowest evaluation, and 5 is the best*); *helpful* – presents user feedback about the quality and helpfulness of the review; *summary* – gives some short texts of the customer's review or subject matter. Only *overall* and *review text* data fields were used in the experiments. An example of the review text is presented below:

```
{"reviewerID": "AUI00LXAB3KKT", "asin": "B004A9SDD8", "reviewerName": "A
Customer", "helpful": [0, 0], "reviewText": "Glad to finally see this app on the
android market. My wife has it on her iPhone and iPad and my son (15 months)
loves it! Hopefully more apps like this are on the way!", "overall": 5.0,
"summary": "Great app!!!", "unixReviewTime": 1301184000, "reviewTime": "03 27,
2011"}.
```

The data consist of different customer reviews given by $D = \{d_1, d_2, d_3 \dots d_n\}$, where n is the total number of reviews. These reviews are classified by different customers, having a certain category assigned to the review with a rating numerical value of $C = \{C_1, C_2, C_i \dots C_5\}$, where C_i ($C_i = i$, where i is a class index), m is the total number of classes ($m = 5$) and considered as a label or class. The data class distribution C_i in the data set is presented in Fig. 1. To improve the classification, it was decided to split the data to equally distributed sets per each class and using the method for measuring the skewness of data (Rennie, 2003), so that each class would collect an equal number of customer product-review records.

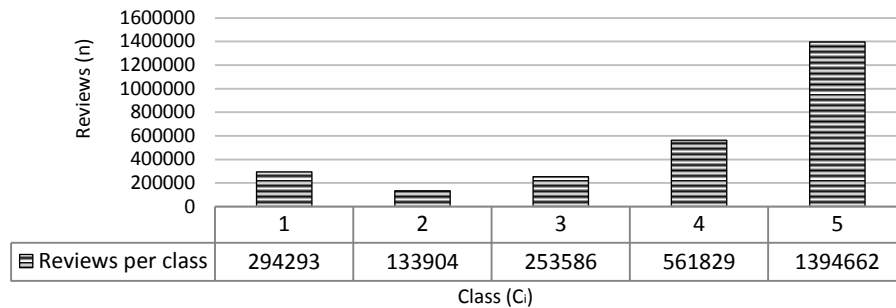


Fig. 1. Distribution of customers' reviews by classes

A workflow model for review processing was established to compare Naïve Bayes, Random Forest, Decision Tree, Support Vector Machines, and Logistic Regression classifiers. However, Joachims (Joachims, 1998) in his comparative work on the text classification with supervised machine learning has concluded that Support Vector Machine is one of the best classifiers, compared to that of Decision Tree or Naïve Bayes. Other authors also demonstrated the superiority of Support Vector Machine over Decision Tree, and Naïve Bayes (Dumais et al., 1998). Later, the Support Vector Machine method was chosen by many researchers and became the most popular method for classifying texts. We decided to make a comparison and include a less investigated Logistic Regression classification method, because it is still used in practical tasks as one of the most accurate classification methods.

Fig. 2 presents the workflow model for review processing that has been used in this research and highlighting the path of the best performed classification method. This workflow model is a modified version of that presented by Seddon (Seddon, 2015). The workflow consists of four key stages:

- **Data extraction.** The main goal of this stage is to select only the required and related data fields to process the data and optimize memory usage. This stage was carried out as follows:
 - Only *overall* and *review text* fields are taken from the input dataset.
 - Collecting the equal number of customer product-review records in each class (i.e. skewness method).
- **Preparation of review texts.** The main goal of this stage is to prepare *review text* fields for extraction of features (Fig. 2). This stage was carried out as follows:
 - Tokenizing each single word by punctuation or white space.
 - Removing all stop words (Stop word corpus was taken from the NLTK website (Natural Language Toolkit Project)), such as *a* and *the*, stop words *a* and *the* have often been in use in any text, but do not include specific information required to train this data model.
 - Putting all the capital letters in a lower case.
 - Stemming (with Porter stemmer) and reducing inflectional forms to a stemma form.

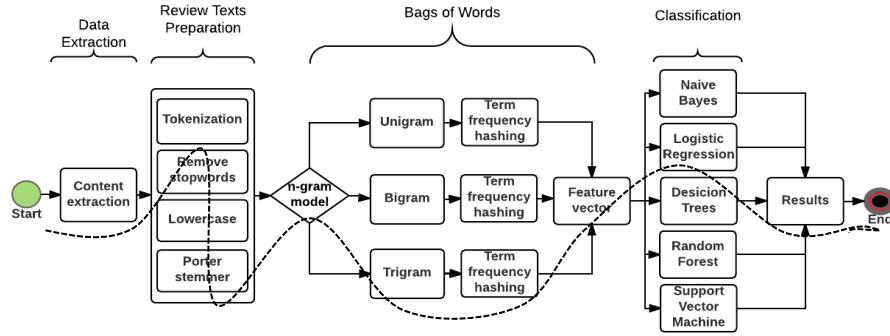


Fig. 2. Workflow model for review processing

- **Bags of words.** The n -gram method as a sequence of written words of length n is applied to construct bags of words. It is a process to split the sentence into words and group them using a combination of n -grams. This stage was carried out as follows:
 - Bags of words (unigrams, bigrams, trigrams) are created from review texts that have passed previous stages, based on the selected n -gram model. Instead of building n -grams from the sentences, continuous text flow is in use. This is because the task of classifier isn't attempting to understand the meaning of a sentence, it basically creates the input to classifier with all features (tokenized terms, and term groups), classifier build the model that assigns the class as accurately as possible.
 - N-gram models might also include more specific properties, using apostrophes, simple word segmentation, phrases, parts of speech, etc.
 - These words are imported to a specially created hashing term-frequency vectorizer that counts the frequency in the set and assigns a unique numerical value for the next classification stage, as well as the weights needed for each word. In other words, a term frequency is identifying how important a word is to a review in a corpus, i.e. the key as a word and value as the number of frequency in the given review set.
 - The feature vector transforms words in to the numerical value represented in the integer format, i.e. the numerical value to the given word and second - the value of frequency of the word.
- **Classification.** This stage was carried out as follows:
 - Data training and testing were performed by the selected classification method using 10-fold cross-validation.
 - Calculating the average classification accuracy for the test data. The average accuracy formula for multi-class classification can be presented as follow (Sokolova and Lapalme, 2009):

$$Accuracy = \frac{\sum_{i=1}^l \frac{tp_i + tn_i}{tp_i + fn_i + fp_i + tn_i}}{l} \times 100\% \quad (1)$$

where tp_i are true positive classification examples, fp_i are false positive ones, fn_i are false negative ones, and tn_i are true negative ones, l is the number of classes. The classification accuracy is calculated by actual labels that are equal to predicted label divided by total corpus size in test data.

The infrastructure of data-processing cluster consists of the master with 4 vCPU and 26 GB of memory and two workers with 2 vCPU, each of them having 13 GB of memory. The infrastructure was provided in the Google Cloud Platform. The experiments were done using Apache Spark v1.6.2, Python v2.7.6 and NLTK v3.0.

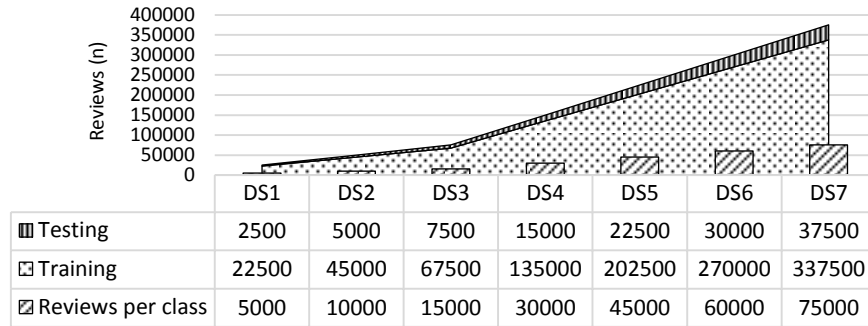


Fig. 3. Composition of data sets for training and testing

Seven data sets DS1, DS2, DS3, DS4, DS5, DS6, DS7 of varied sizes were used in our experiments. Composition of a data set for training and testing is distributed like this: 90% for training and 10% for testing, and the equal number of reviews per class (Fig. 3).

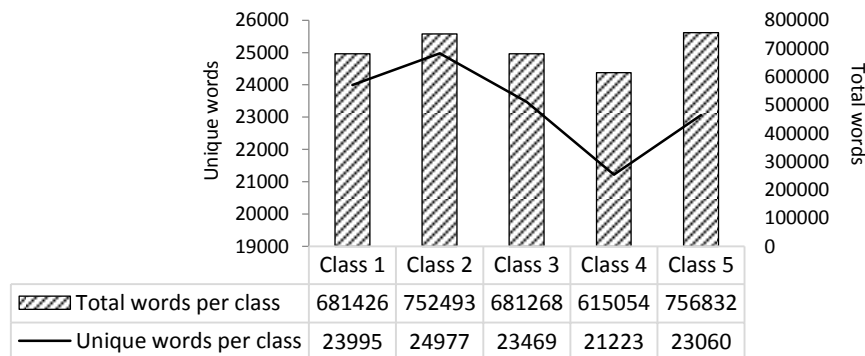


Fig. 4. Total and unique words per class (terms)

Fig. 4 presents the statistics of the unique and total words (terms). All unique words are counted in comparison of all the words existing in the given data set per each class. In general, the selected text corpus has unique words that consist of less than 10% of total words and distribution of unique words has higher values in class 2 and lower in class 4. Usually, the unique words represent the given class very well, and reasonable similarities exist between 1 and 2, 4 and 5 classes.

3 Evaluation of the classification experiment

Fig. 5 – Fig. 9 illustrate the comparison of classification accuracy of multinomial Naïve Bayes, Random Forest, Decision Tree, Support Vector Machines with the linear kernel

and Stochastic Gradient Descent optimization algorithm (Gupta et al., 2014), and Logistic Regression with limited memory Broyden–Fletcher–Goldfarb–Shanno optimization algorithm (Mokhtari et al., 2015) classification methods related to the classification accuracy, the number of product reviews, and combination of n -grams. The classification methods were used with their default parameters that are configured in Spark v1.6.2 MLlib library, except the number of features, trees and depth – these were customized according to the size of the data and limitations associated with the use of computing resources.

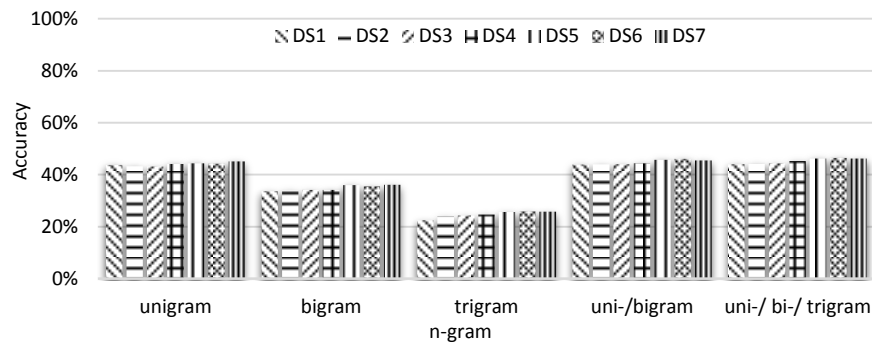


Fig. 5. Classification accuracy of Naïve Bayes

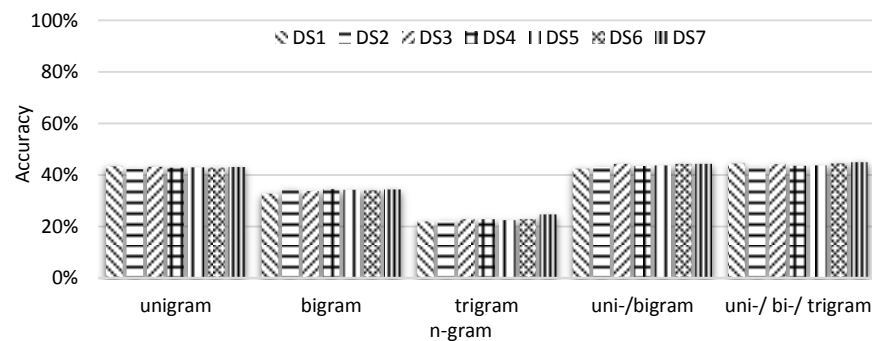


Fig. 6. Classification accuracy of Support Vector Machine

Support Vector Machine with the linear kernel is very fast method but it doesn't always give the best classification accuracy comparing to Support Vector Machine with the non-linear kernels. Training process of Support Vector Machine with the non-linear kernels is hard to distribute, and therefore, these methods are not yet implemented in the Apache Spark machine learning library.

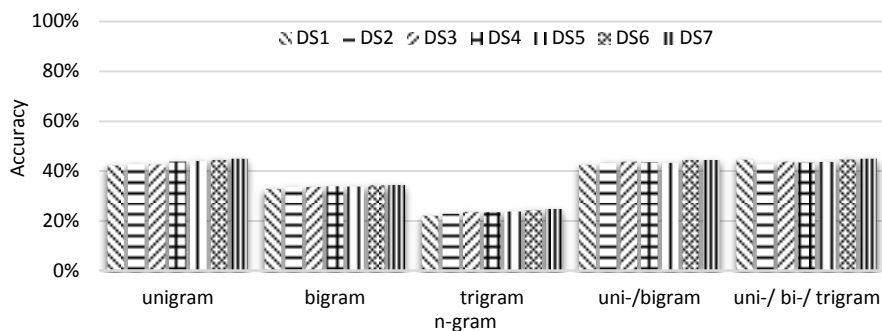


Fig. 7. Classification accuracy of Random Forest

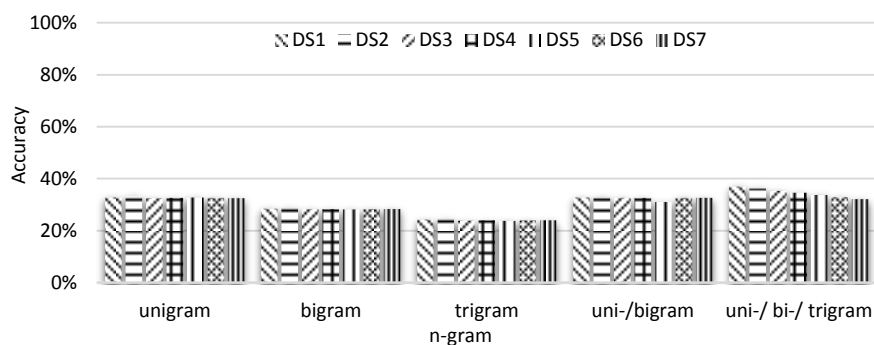


Fig. 8. Classification accuracy of Decision Tree

The results of Decision Tree classification accuracy were lowest (min in trigram: 24.10%, max in uni/bi/tri-gram: 34.58%) as compared to the classifiers analyzed.

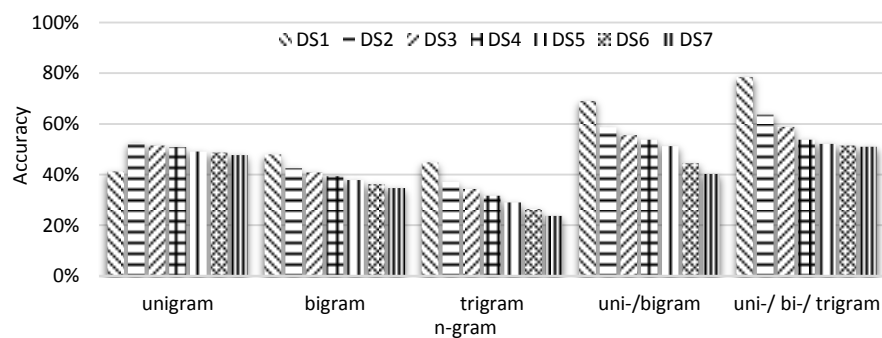


Fig. 9. Classification accuracy of Logistic Regression

The findings indicate that the Logistic Regression multi-class classification method with the given data of product-reviews is the best (min 32.43%, max 58.50%) classification accuracy in comparison to the analyzed classifiers. Logistic Regression multi-class classification method is less stable method as the values of average classification accuracy are sparsely distributed in comparison to other methods.

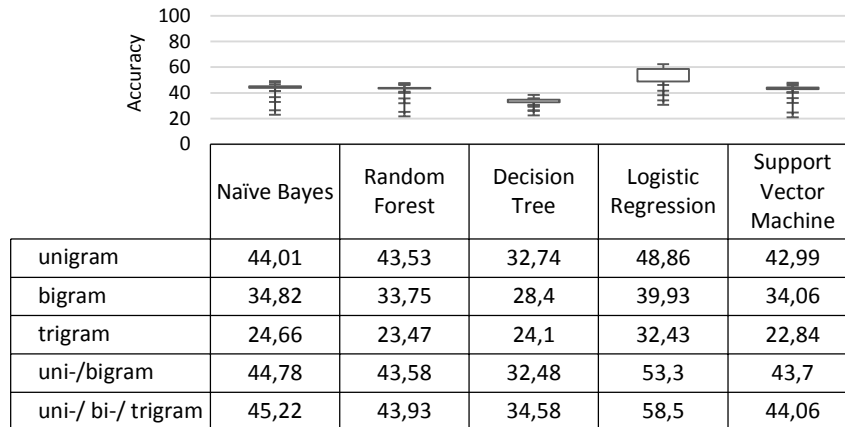


Fig. 10. Average classification accuracy

Fig. 10 illustrates that the average values of classification accuracy of Naïve Bayes, Random Forest, and Support Vector Machine are similar (min in trigram: 33 – 34%, max in uni/bi/tri-gram: 43 – 45%), and Naïve Bayes has achieved 1 – 2% higher average classification accuracy results in comparison to Random Forest and Support Vector Machine, but the difference is not statistically significant. Except Logistic Regression, performance of analyzed classification methods contains more stability and the values of the average classification accuracy are less distributed.

4 Conclusions

The comparison of Naïve Bayes, Random Forest, Decision Tree, Support Vector Machines, and Logistic Regression methods for multi-class text classification is presented in this paper.

The findings indicate that the Logistic Regression multi-class classification method for product-reviews has achieved the highest (min 32.43%, max 58.50%) classification accuracy in comparison with Naïve Bayes, Random Forest, Decision Tree, and Support Vector Machines classification methods. On the contrary, Decision Tree has got the lowest average accuracy values (min in trigram: 24.10%, max in uni/bi/tri-gram: 34.58%).

The experimental results have shown that the Naïve Bayes classification method for product-review data achieves 1 – 2% higher average of classification accuracy than the

Random Forest and Support Vector Machine method, but the difference is not statistically significant.

Following the comparative analysis, it can be indicated that the overall classification accuracy in combination with uni/bi/tri-gram models increases the average of classification accuracy, but these values are insignificant as compared with the unigram model of all classification methods.

The investigation indicates that increasing the size of the training data set from 5000 to 75000 reviews per class leads to insignificant growth of the classification accuracy (1 – 2%) of Naïve Bayes, Random Forest, and Support Vector Machines classifiers. These results show that a training set size of 5000 reviews per class is sufficient for all analyzed classification methods, and classification accuracy relates more to the n -gram properties.

References

- Agrawal D. , Bernstein P. , Bertino E., Davidson S., Dayal U. (2011). Challenges and Opportunities with Big Data. Cyber Center, Purdue University. West Lafayette : Purdue e-Pubs, (2011). pp. 8 - 9, Technical Report.
- Agrawal R., Gupta A., Prabhu Y., Varma M. (2013). Multi-Label Learning with Millions of Labels: Recommending Advertiser Bid Phrases for Web Pages. WWW '13 Proceedings of the 22nd international conference on World Wide Web. (2013), pp. 13 - 24.
- Alpaydin E. (2010). Introduction to Machine Learning. Second Edition. Cambridge : The MIT Press, (2010). pp. 1 - 3. ISBN-13: 978-0262012430.
- Bi W., Kwok J.T. (2013). Efficient Multi-label Classification with Many Labels. [ed.] Sanjoy Dasgupta and David McAllester. Proceedings of the 30th International Conference on Machine Learning (ICML-13). (2013), Vol. vol. 28, pp. 405-413.
- Breaking the Spherical and Chromatic Aberration Barrier in Transmission Electron Microscopy. Freitag B., Kujawa S., Mul P., Ringnalda J., Tiemeijer P. (2005). 3, (2005), Ultramicroscopy, Vol. 102, pp. 209–214.
- Cambria E., White B. (2014). Jumping NLP Curves: A Review of Natural Language Processing Research. s.l. : IEEE, (2014), Vol. 9, pp. 48 - 57.
- Caraciolo M. (2011). Machine Learning with Python - Logistic Regression. [Online] ARTIFICIAL INTELLIGENCE IN MOTION, (2011). [Cited: September 5, 2016.] <http://aimotion.blogspot.lt/2011/11/machine-learning-with-python-logistic.html>.
- Cavnar W.B., Trenkle J. M. (1994). N-Gram-Based Text Categorization. (1994).
- Cecchetto B. T. (2014). Correction of Chromatic Aberration from a Single Image Using Keypoints. (2014).
- Cheng J. , Mizil C. D. N. , Leskovec J. (2014). Antisocial Behavior in Online Discussion Communities. [Online] (2014). <http://cs.stanford.edu/people/jure/pubs/trolls-icwsm15.pdf>.
- Christiansen M., Chater N. (2003). Language evolution: The hardest problem in science? Language Evolution. (2003), pp. 1 - 15.
- Chu C.-T., Kim S. K., Lin Yi-An, Yu Y.Y., Bradski G., Ng A. Y., Olukotun K. (2007). Map-Reduce for Machine Learning on Multicore. (2007).
- Chung S.-W., Kim B-K, Song W.-J. (2010). Removing Chromatic Aberration by Digital Image Processing. June (2010), Vol. 49, 6.
- Daudaravičius V. (2012). Collocation segmentation for text chunking. Kaunas : Vytautas Magnus University, (2012).
- Dumais S., Platt J., Heckerman D., Sahami M. (1998). Inductive learning algorithms and representations for text categorization. (1998), pp. pp. 148–155.

- Dzemyda G., Kurasova O., Žilinskas J. (2008). Daugiamačių duomenų vizualizavimo metodai. Vilnius : Mokslo aidai, (2008). pp. 10 - 12. ISBN 978-9986-680-42-0.
- Flannery B. P., Teukolsky S., Press W. H., Vetterling W. T. (2007). Section 16.5. Support Vector Machines. Numerical Recipes: The Art of Scientific Computing. 3rd. New York : Cambridge University Press, (2007).
- Frakes W., Baeza-Yates R. (1992). Information Retrieval: Data Structures and Algorithms. Chapter 8: Stemming Algorithms. s.l. : Prentice Hall, (1992). p. Information Retrieval: Data Structures and Algorithms . 0134638379.
- Grefenstette G., Tapanainen P. (1994). What is a Word, what is a Sentence? Problems of Tokenization. (1994).
- Gross, Herbert and Blechinger, Fritz. (2007). Aberration Theory and Correction of Optical Systems. Handbook of Optical Systems. s.l. : Wiley VCH, (2007), Vol. 3.
- Gu L., Li H. (2013). Memory or Time: Performance Evaluation for Iterative Operation on Hadoop and Spark. High Performance Computing and Communications & 2013 IEEE International Conference. (2013), pp. 725-727.
- Gupta M. R., Bengio S., Weston J. (2014). Training Highly Multiclass Classifiers. [ed.] Koby Crammer. Journal of Machine Learning Research. (2014), Vol. 15.
- Hastie T., Tibshirani R., Friedman J. (2009). The Elements of Statistical Learning. Second Edition. New York : Springer-Verlag, (2009). ISBN 978-0-387-84857-0.
- Ivanov K. (1972). Quality-control of information: On the concept of accuracy of information in data banks and in management information systems. Stockholm, Sweden : The Royal Institute of Technology KTH, December 11, (1972). Doctoral dissertation.
- Joachims, T. (1998). Text categorization with support vector machines: learning with many relevant features. (1998), pp. 137-142.
- Kang S. B. (2007). Automatic Removal of Chromatic Aberration from a Single Image. (2007), pp. 1 - 8.
- Karau H., Konwinski A., Wendell P., Zaharia M. (2015). Learning Spark. s.l. : O'Reilly Media, Inc, (2015). 978-1-449-35862-4.
- Karau, Holden, et al. (2015). Learning Spark. Sebastopol : O'Reilly Media, Inc., (2015). 978-1-449-35862-4.
- Kidger M. J. (1997). The Importance of Aberration Theory in Understanding Lens Design. (1997), Vol. 3190, pp. 26-33.
- Kozubek M., Matula P. (2000). An Efficient Algorithm for Measurement and Correction of Chromatic Aberrations in Fluorescence Microscopy. December (2000), Vol. 200, 3, pp. 206-217.
- Lanford J., Nykodym T., Rao A., Wang A. (2015). Generalized Linear Modeling with H2O's R. s.l. : H2O.ai, (2015).
- Leskovec J., Krevl A. Stanford university. Stanford Large Network Dataset Collection. [Online] [Cited: March 9, 2016.] <http://snap.stanford.edu/data/>.
- Manning Ch.D., Raghavan P., Schütze H. (2008). Introduction to Information Retrieval. Online Edition. Cambridge : Cambridge University Press, (2008). p. 258. ISBN: 0521865719.
- McAuley J., Pandey R., Leskovec J. (2015). Inferring networks of substitutable and complementary products. Knowledge Discovery and Data Mining. (2015).
- McAuley J., Targett C., Shi J., Hengel A. (2015). Image-based recommendations on styles and substitutes. SIGIR. (2015).
- Mell P., Grance T. (2011). The NIST Definition of Cloud Computing. Gaithersburg, : U.S. Department of Commerce, (2011).
- Meng X., Bradley J., Yavuz B., Venkataraman S., Liu D., Freeman J., Tsai D. B., Amdie M, Owen S., Xin D., Xin R., Franklin M. J., Zadeh R., Zaharia M., Talwalkar A. (2015). MLlib: Machine Learning in Apache Spark. May 26, (2015).
- Mokhtari A., Ribeiro A. (2015). Global Convergence of Online Limited Memory BFGS. [ed.] L'eon Bottou. Philadelphia : arXiv owned Cornell University, December (2015). Vol. 16.
- More J. J. (1977). The Levenberg-Marquardt Algorithm: Implementation and Theory. (1977), pp. 105-116.

- Natural Language Toolkit Project. Natural Language Toolkit. Natural Language Toolkit. [Online] [Cited: March 1, (2016).] <http://www.nltk.org/>.
- Ouda K. (2015). Distributed Machine Learning: A Review of current progress. (2015).
- Philip Chen C.L., Zhang C.-Y. (2014). Data-intensive applications, challenges, techniques and technologies: A survey on Big Data. *Information Sciences*. (2014), Vol. 275, pp. 314–347.
- Phyu T.N. (2009). Survey of Classification Techniques in Data Mining. *Proceedings of the International MultiConference of Engineers and Computer Scientists 2009*, (2009), Vol. Vol. 1.
- Pöntinen P. (2012). Study on Chromatic Aberration of two Fisheye Lenses. (2012), Vol. XXXVII, p. 27.
- Rennie M., Shih L., Teevan J., Karger D. (2003). Tackling the Poor Assumptions of Naive Bayes Text Classifiers. (2003).
- Rokach L. , Maimon O. (2005). Top-down induction of decision trees classifiers-a survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*. November (2005), Vol. 35, 4, pp. 476–487.
- Rudakova V., Monasse P. (2013). Precise Correction of Lateral Chromatic Aberration in Images. September (2013), Vol. 8333, pp. 12-22.
- Seddon M. (2015). Natural Language Processing with Apache Spark ML and Amazon Reviews. [Online] (2015). [Cited: March 10, 2016.] <https://mike.seddon.ca/natural-language-processing-with-apache-spark-ml-and-amazon-reviews-part-1/>.
- Soares J. V. B., Leandro J. J. G., Cesar R., Jelinek H. F., Cree M. J. (2006). Retinal Vessel Segmentation Using the 2-D Gabor. September (2006), Vol. 25, pp. 1214-22.
- Sokolova M., Lapalme G. (2009). A systematic analysis of performance measures for classification tasks. (2009), Vol. 45, pp. 427–437.
- Willson R., Shafer S. (1991). Active Lens Control for High Precision Computer Imaging. April (1991), Vol. 3, pp. 2063 - 2070.
- Wu X., Kumar V., Quinlan J. Ross, Ghosh J., Yang Q., Motoda H., Motoda H., McLachlan G. J. , Ng A., Liu B., Yu P. S. , Zhou Z.-H., Steinbach M, Hand D. J., Steinberg D. (2008). Top 10 algorithms in data mining. (2008).
- Yang, Y, Huang S., Rao N. (2008). An Automatic Hybrid Method for Retinal Blood Vessel Extraction. September (2008), Vol. 18, 3, pp. 399-407.
- Zana F., Klein J.-C. (2001). Segmentation of vessel-like patterns using mathematical morphology and curvature evaluation. July (2001), Vol. 10, pp. 1010 - 1019.
- Zhang Y, Jin R., Zhou ZH. (2010). Understanding Bag-of-Words Model: A Statistical Framework. *International Journal of Machine Learning and Cybernetics*. December (2010), Vol. Volume 1, 1, pp. pp 43–52.

Received January 28, 2017, revised June 12, 2017, accepted June 14, 2017