

Sentiment Polarity Detection in Azerbaijani Social News Articles

Sevda Mammadli¹, Shamsaddin Huseynov¹, Huseyn Alkaramov¹, Ulviyya Jafarli¹,
Umidsuleymanov², Samir Rustamov^{1,3}

¹ School of Information Technologies and Engineering, ADA University, Baku, Azerbaijan

² E-GOV Development Center, Baku, Azerbaijan, ³ Institute of Control Systems, Azerbaijan

{smammadli2019, shuseynov2019, halkaromov2019, ujafarli2019}@ada.edu.az

umidsuleymanov96@gmail.com

srustamov@ada.edu.az

Abstract

Text classification field of natural language processing has been experiencing remarkable growth in recent years. Especially, sentiment analysis has received a considerable attention from both industry and research community. However, only a few research examples exist for Azerbaijani language. The main objective of this research is to apply various machine learning algorithms for determining the sentiment of news articles in Azerbaijani language. Approximately, 30.000 social news articles have been collected from online news sites and labeled manually as negative or positive according to their sentiment categories. Initially, text preprocessing was implemented to data in order to eliminate the noise. Secondly, to convert text to a more machine-readable form, BOW (bag of words) model has been applied. More specifically, two methodologies of BOW model, which are tf-idf and frequency-based model have been used as vectorization methods. Additionally, SVM, Random Forest, and Naive Bayes algorithms have been applied as the classification algorithms, and their combinations with two vectorization approaches have been tested and analyzed. Experimental results indicate that SVM outperforms other classification algorithms.

1 Introduction

Development of technology generates a vast amount of data flow through the internet and encourages the creation of sophisticated methodologies to store and analyze it. Analyzing such huge volumes of data manually could lead to the waste of time and investment. Therefore, currently more

automated and efficient ways are implemented to solve the problem.

With the growth of the produced data, a branch of Artificial Intelligence - Natural Language Processing (NLP) had begun to evolve. Text classification is a fundamental part of NLP and has been applied in many areas. The aim of text classification is to group data into predefined categories based on the labeled data using various machine learning techniques. It requires several stages to categorize the data including data collection, preprocessing, and feature extraction. One of the implementation areas of text classification includes sentiment analysis. Sentiment analysis helps to define whether an author's opinion towards a specific topic is negative or positive. Considering the fact that people's opinion directly influences the businesses and organizations it is no surprise that sentiment analysis receives a lot of attention.

Although sentiment analysis has been applied largely worldwide, research on its usage and utility on Azerbaijani language is scarce. English language has the luxury of having numerous annotated datasets as well as having well-tuned text preprocessing techniques. Different from English language, natural language processing algorithms are not improved sufficiently in Azerbaijani which is an agglutinative language and therefore requires special pre-processing approaches. Furthermore, implementation of some well-known feature extraction approaches is not experimented enough and investigating their effectiveness in natural language processing tasks namely, in sentiment analysis for an agglutinative language is one of the main objectives of the research. Additionally, the main purpose of our research is to build supervised machine learning based, automatic sentiment polarity detection system for analyzing Azerbaijani social news articles.

2 Literature Review

As sentiment analysis is one of the prominent topics nowadays there exists vast amount of experiments applied with different methods. Sentiment analysis of text for Azerbaijani language had been investigated by Aida-zade et al. (2013). Multi machine learning algorithms had been applied for news classification in Azerbaijani language in (Suleymanov and Rustamov, 2018; Suleymanov et al., 2018; Aida-zade et al., 2018). Cambria (2016) distinguishes three main approaches in the field of sentiment analysis: knowledge-based, statistical and hybrid. The first one is the method of classifying text using rule-based algorithm to extract the sentiment. To help the organizations to improve their decision making and improve customer satisfaction Zaw and Tandayya (2018) applied rule-based algorithm called Contrast Rule-Based sentiment analysis to classify customer reviews automatically. Another rule-based algorithm is proposed by Tan et al. (2015) for classifying financial news articles. According to the research, initial stage was to determine the sentiment of each single sentence in the given financial news. Next stage was calculating positivity/negativity for the whole content of an article.

Sentiment analysis is widely used to measure the public opinion about a given topic. Li et al. (2017) investigated relationship between Dow Jones Industrial Average and public emotions. During experiment approximately 200 million twitter data was collected that mentions 30 companies which are part of the New York Stock Exchange. Researchers applied a different methodology called SMeDA-SA. The method initially extracts all uncertain sentences from the document to create the vocabulary. As a result, the research indicated that stock price of companies can be predicted with the given methodology.

One of sentiment analysis task – subjectivity detection applied before sentiment analysis for increasing accuracy performance. Rustamov et al. applied Adaptive Neuro Fuzzy Inference System (2013a), Hidden Markov Models (2013b) and Hybrid models (2018) for detection of subjectivity analysis. Same techniques had been applied for document level sentiment analysis (Rustamov et al., 2013). The method described by Araque et al. (2017) is an example of deep learning algorithm usage for sentiment analysis. Recently, deep learning is widely used to classify the text as it is capa-

ble of extracting public opinion regarding a specific topic and also works excellently with high-level features. During the research, deep learning model was developed using word embedding and linear machine learning model was implemented.

Sentiment analysis can also be used with the combination of different methodologies in the implementation of various applications. One of the researches that benefited from application of sentiment analysis is done by Rosa et al. (2019). In the research, applications collect data about a user and give recommendations based on the data produced by the user. The research proposes an approach for a recommendation system which takes user's current psychological state into account using sentiment analysis. Based on the mood of a user system sends different messages to the user including relaxing, peaceful, calm and etc.

Bansal and Srivastava (2018) applied word2vec model with machine learning algorithms to classify user reviews. The word2vec model was used to represent 400.000 consumer reviews data from Amazon as vectors. Later, the vector representation of data was given to the classifier as an input. In order to classify the data both Continuous Bag of Words (CBOW) and Skip-Gram model were implemented in combination with 4 machine learning algorithms including SVM, Naive Bayes, random forest, and logistic regression.

Severyn and Moschitti (2015) worked on an application that does sentiment analysis of tweets with deep learning models. They have applied unsupervised natural language model to initialize word embeddings which were used as distant supervised corpus in their deep learning model. The model was initialized by using pre-trained parameters of network, then trained on supervised training data from Semeval-2015. According to official test sets' result, their model ranked first in phrase-level and second in message-level tasks.

In some researches it is observed that dictionary-based approaches are also effective to extract the sentiment from text data. Nigam and Yadav (2018) divided collected tweets into lexemes and matched the words with the terms in the dictionary. Matched words were weighted so that negative word gets -1 score and positive word gets +1 score. The overall sentiment of document was calculated by subtracting the weights of positive words from weights of negative words.

3 Methodology

3.1 Data Collection

Data collection and annotation is an essential step for training supervised machine learning algorithms. Not only the collected data should be relevant to the research objective but also the annotation process should be carefully designed in order to have no inconsistencies among labeled data as they could scale up during training phase and lead to unsatisfactory results. For conducting this research using supervised machine learning techniques, approximately 30000 news articles had been collected and monitored from online websites of famous Azerbaijani newspapers. News articles under social category have been observed to contain more sentiment polarity and therefore found to be more suitable for sentiment analysis. The inter-annotator agreement had been established and an additional procedure had been implemented as a control mechanism in order to verify that the agreement had been followed during annotation. One of the main requirements of agreement was to label only the articles in which sentiment has been explicitly expressed. For instance, an article about a social event cannot be annotated for its sentiment unless author explicitly shows its social benefits or downsides. This was done to eliminate any inconsistencies emerging from subjective assessment of annotators. The control mechanism had been implemented as follows. Firstly, each news article was given a unique identifier and after random shuffling, articles were divided into small chunks each containing 500 articles. In the first run each annotator was given a chunk. After all annotators finished the first chunk, the second run began. In the second run, each annotator was given a chunk and additionally 50 more already labeled articles without their labels. By comparing these 50 articles' new labels with old ones, we could determine in which aspects annotators did not agree and made relevant adjustment to the annotation agreement to minimize the amount of disagreement in subsequent runs. In the following runs the steps of the second run were repeated until there was no chunk remaining. 12210 articles were labelled according to above mentioned rules. Among the labeled news articles, 4565 of them were labeled as positive and 7645 were labeled as negative. In the next stages, we had applied k-folds cross validation. In this method k stands for the number of repetitions of splitting data into test and train part. Cross validation is a method in machine learning that is used for

assessing the result of the applied algorithm. It helped us to estimate how accurately the model would work in real case situations. By considering that we may not have satisfactory amount of data to train the model, and while splitting data into test and train we eliminate some of the train data and it may cause underfitting problem, we have applied 10-fold cross validation. After each splitting we got the accuracy score and when the splitting ended, we obtained final accuracy by calculating the average.

3.2 Data Preprocessing

Getting clean data was first step of the experiment. In order to get sentiment from data, stop words were cleaned, which have no sentiment but are highly frequent in the dataset and could decrease accuracy. Especially while applying frequency based vectorizer, noisiness of data interrupts quality of classification process.

Dataset contained XML and HTML tags since they were taken from online resources. Especially names of websites, sources of the article, dates, URL links and JS tags were present in the dataset and they affected the prediction accuracy negatively. For example, website names which end with the domain names such as “.az”, “.com”, “.org”, “.net”, “.ru”, “.edu”, “.gov” had been removed. In addition, the ones that started with “http”, “https” and “www” and extra time tags were also deleted from data. Furthermore, all unnecessary punctuations were cleaned except semicolon and dash, since they are used in compound words in Azerbaijani language.

Finally, to eliminate difference between same words with different cases (uppercase and lowercase), all tokens had been converted to lowercase. It is needed to mention that after preprocessing, number of words decreased, and consequently, dictionary size was reduced as well which speeded up the processing of data and the classification algorithms.

3.3 Feature Extraction

In terms of natural language processing, there is an essential need to convert text data into a specific format which is appropriate for applying statistical machine learning algorithms. The process is called feature extraction and there exists different methodologies for feature extraction. One of the commonly used methods is called bag of words model (BOW) that treats each single word as a feature. This approach takes collection of docu-

ments and converts it into a list of unordered words called vocabulary (Chen et al., 2017). Next step is to create a vector representation of documents according to the size of given vocabulary where existence of each unique word defines the size. In a basic vectorizing models, it counts occurrence of each word in text and converts it to an array of real numbers.

In the domain of machine learning, there exists various vectorization approaches one of which is counting based. Also called frequency based, it provides a sparse representation of corpus of documents as a matrix. However, frequency based vectorizer has several drawbacks. Firstly, not all words have sentiment value despite how frequently it is used in the document, namely, frequency of commonly used words could shadow other more significant and sentiment containing words. Therefore, to solve this problem term-frequency and inverse-document-frequency method (tf-idf) is widely used. In addition to the number of occurrence in the document tf-idf also takes into consideration word's density in the whole corpus of documents. It consists of two parts where term frequency provides count of each word and inverse document frequency reduces value of words that are densely used in the corpus.

Another approach of bag of words model is hashing based vectorizer. It maintains vectors representing each term as an integer value. Different from others, it does not create dictionary, still having larger matrix to encode a document. tf-idf and frequency based vectorizer generates high-dimensional vector representations. Unlike them hashing based vectorizer suggests an efficient way to reduce the dimensionality of the vector. It offers default size for feature vector and provides an option to reduce and increase vector size. However, probability of collision should be considered when choosing the size. If the number of unique words in the vocabulary exceeds the feature size it could lead to collision where several unique words could map to the same integer value.

It is not necessary to present each single unique term as an input to the vectorization method. Different word combinations can be used here including unigram, bigrams, and trigrams. In unigram each word represents one feature. Additionally, we can also take combination of two words called bigrams, and combination of three words at a time called trigrams (Bhavitha et al., 2017).

4 Classifiers

4.1 Random Forest

Random forest is one of the supervised learning algorithms, which is implemented in both regression and classification problems. This classifier is a collection of recursive, tree structured models.

In decision tree, the prediction is done by splitting root training set into subsets as nodes, and each node contains output of the decision, label or condition. After sequentially choosing alternative decisions, each node recursively is split again and at the end classifier defines some rules to predict result. Conversely, in random forest, classifier randomly generate tree without defining rules.

We have implemented random forest algorithm with the different vectorizer methodologies, and n-gram models as shown in Table 1 and got different outcomes as described below.

Feature extraction	n-grams	F1-Score	Precision	Recall
Frequency based vectorizer	Unigram	93.21	91.02	95.87
	Bigram	92.15	88.47	95.97
	Trigram	89.79	82.91	97.38
Tf-idf	Unigram	93.33	90.65	95.93
	Bigram	92.27	88.96	96.61
	Trigram	89.95	83.29	97.5

Table 1: Random Forest Classifier Result

While considering the highest F1 score, we got 93.33 percent from the combination of tf-idf vectorizer and unigram model. On the other hand, when taking highest recall and precision score separately, we obtain the highest recall score of 97.38 percent. The highest recall score yields the precision of 82.91 and F1 score of 89.79 percent. Even though the recall is the highest, we got the lowest F1 score from that combination. Additionally, the highest precision score which is 91.02 percent provides 93.21 F1 score and the lowest recall score of 95.87 percent. As visible in the Table 1 the second highest F1 score of 93.21 was ob-

tained from the combination of frequency based vectorizer and unigram model. It is noticeable that the difference between the first highest F1 score of 93.33 percent and the second highest F1 score of 93.21 is too close to each other and therefore was not statistically important. The lowest F1 score of 89.79 percent was obtained from the combination of frequency based vectorizer and trigram model.

4.2 Naïve Bayes

The second machine learning algorithm we have applied to our data set is Naive Bayes classifier. It is a probabilistic model, which is derived from Bayes Theorem that finds the probability of hypothesis activity to the given evidence activity. According to naive Bayes rule each feature is independent from each other and because of the assumption about independence, occurrence of one feature has no impact to others. Depending on the features, Bayes classifier has several forms including Gaussian Naive Bayes classifier, Multinomial Naive Bayes, and Bernoulli Naive Bayes. In this research Multinomial Naive Bayes which is mostly used for document level analysis is implemented. In this classifier, feature representation has been generated by multinomial distribution which reflects frequency of words like vectorization. Probability of an event i happening with multinomial Naive Bayes is formulated as:

$$p(x|C_k) = \frac{(\sum x_i)!}{(\prod x_i)!} \prod_i p_k i^{x_i}$$

In this paper we have applied Multinomial Naive Bayes classifier using $\alpha=1$ with frequency based vectorizer and tf-idf vectorizer as described in the following table. Alpha parameter prevents model assigning null probabilities in case of 0 term frequency. Below, table 2 clearly indicates the impact of combination of different vectorization methodologies, n-gram models on the F1, recall, and precision score.

Table 2 presents that combination of frequency based vectorizer and bigram model provided the highest F1 score result of 95.47 percent. In contrast, the lowest F1 score result we got among the above combinations was 90.9 percent and was obtained from tf-idf and trigram model. As can be seen from the table the lowest F1 score yields the lowest precision score of 90.0 percent. The highest precision score we got was 97.87 which yields the lowest recall score of 85.4 percent. The high-

est F1 score of 95.47 percent was obtained from the combination of frequency based vectorizer and bigram model. Additionally, it is noticeable from the table that while considering tf-idf results the highest F1 score we got 94.66 percent which was obtained from bigram model. This combination yields the recall score of 98.24 percent which is the second highest recall score and precision score of 91.34 which is the lowest precision score.

Feature extraction	n-grams	F1-Score	Precision	Recall
Frequency based vectorizer	Unigram	94.87	95.87	93.9
	Bigram	95.47	97.24	93.77
	Trigram	91.2	97.87	85.4
Tf-idf	Unigram	94.1	91.54	96.97
	Bigram	94.66	91.34	98.24
	Trigram	90.9	84.2	98.75

Table 2: Multinomial Naive Bayes Classifier Result

4.3 Support Vector Machine

The third classification algorithm that we have applied is SVM. the purpose of SVM classification algorithm is to define optimal hyperplane in N dimensional space to separate the data points from each other. N dimensional space here is number of features:

$$h(x_i) = \text{sign}(w * x_i + b)$$

$$\min_w \lambda \|w\|^2 + \sum_{i=1}^n (1 - y_i < x_i, w >) \quad (1)$$

Equation (1) describes the calculation of cost function and hypothesis for SVM. One of the important terms used in SVM is kernel parameter. When the data is so huge and hardly computational, kernel is used to speed up and optimize SVM. There are different types of the kernel parameter such as linear kernel, polynomial kernel, rbf kernel, and sigmoid kernel. In this research linear kernel parameter is used. In the research, we analyzed accuracy of SVM classification algorithm

with different vectorization and n-gram models which is described in the following table.

Feature extraction	n-grams	F1-Score	Precision	Recall
Frequency based vectorizer	Unigram	95.51	95.4	95.63
	Bigram	95.45	94.17	96.76
	Trigram	92.82	88.73	97.31
Tf-idf	Unigram	96.79	96.48	97.1
	Bigram	95.9	94.45	97.41
	Trigram	93.35	89.19	97.93

Table 3: Linear SVM Result

Table 3 depicts the F1, recall, and precision results for SVM classifier using various feature selection and n-grams models. As can be seen from the table the best F1 score we got 96.79 percent was from the combination of tf-idf vectorizer and unigram model. While considering frequency based vectorizer the highest F1 score which is 95.51 percent was obtained from unigram model. When comparing the F1 score of unigram and bigram models we do not observe huge difference between results. Recall values for the SVM classifier ranged from 95.63 percent to 97.93 percent. The highest recall score was 97.93 which was obtained from the combination of tf-idf and trigram model. As described in table the lowest F1 score 92.82 percent and was gained from trigram model and frequency-based vectorization.

4.4 Data Skewness and Classifier Comparisons

As described in data collection section, the dataset was skewed towards negative samples with 7.6k negative samples and 4.5k positive samples. Having skewness in a dataset can be considered normal as dataset is formed as a result of some natural phenomena which can inherently be biased towards some category or other. In our case, news agencies play the role of data source which seems to be biased towards generating negative news articles. This could be explained by the fact that this

type of articles is catchier and preferred more by the readers.

Having data skewness can have a direct impact on the results of a machine learning model and gaining further insights can contribute to obtaining higher results. Therefore, the impact of skewness on our dataset had been researched. Firstly, we examined the precision and recall score per class to see if skewness has a significant impact on the results.

Class	Recall	Precision
Positive	93.58	94.82
Negative	96.89	96.23

Table 4: Per class precision and recall scores of SVM classifier

Table 4 demonstrates that skewness in the dataset affects the performance of the classifier. Therefore, we explored several approaches for optimizing the results while working with skewed dataset. The considerable change came from the application of adjusting sample weights inversely to the frequencies of each class in SVM classifier. By this way, the classifier is penalized more for the mistakes made on samples from underrepresented class, namely positive. This allowed to elevate the recall score to 95.14 percent for the positive class and increase the precision score to 97.07 percent for the negative class, while slightly lowering the recall to 95.95 percent.

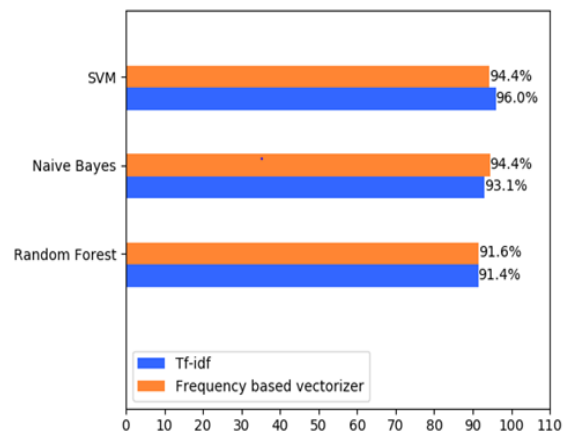


Figure 1: Accuracy of SVM, Naive Bayes, random forest classifiers with vectorization methods

Figure 1 demonstrates the accuracy results obtained from implementation of tf-idf and frequency-based vectorization methods with each classification algorithm. As can be seen from the figure, combination of SVM and tf-idf outperforms other classifiers with the 96 percent accuracy. The minimum accuracy was obtained from the combination of random forest classifier and tf-idf vectorizer with 91.4 percent accuracy. While considering accuracy result it is observable that accuracy range changes between 91 percent and 96 percent. Random forest classifier does not offer huge difference between two vectorization methods as there is only 0.2 percent accuracy difference between tf-idf and frequency-based vectorization methods. When analyzing Naive Bayes classifier, the best accuracy we got was 94.4 percent and was obtained from the frequency based vectorizer.

5 Conclusion

In conclusion, we experimented with three machine learning algorithms on news articles in Azerbaijani language. Comparing one classifier with another, depending on the n-gram model and vectorization method we obtained different results for different combinations. According to the research the highest F1-score we got was 96.79 percent with the implementation of SVM on tf-idf vectorizer and unigram model. Research also revealed that Naive Bayes classifier give its best result 95.47 percent with the combination of frequency based vectorizer and bigram model, while random forest acquires highest F1-score 93.33 percent by using tf-idf based feature extraction and unigram model.

Additionally, for future work we plan to improve our research by enlarging our dataset, adding neutral class, applying rule-based algorithms and observing their performance in our dataset. In addition to them, we are going to apply word embedding with different classification algorithms.

Acknowledgments

This work has been carried out in Center for Data Analytics Research at ADA University in partnership with E-GOV Development Center.

References

Aida-zade Kamil, Samir Rustamov, Mark A. Clements and Elshan Mustafayev. 2018. *Adaptive Neuro-Fuzzy Inference System for Classification of Texts*.

In Zadeh L., Yager R., Shahbazova S., Reformat M., Kreinovich V. (eds) *Recent Developments and the New Direction in Soft-Computing Foundations and Applications. Studies in Fuzziness and Soft Computing*. Springer, 361(2018):63-70. https://doi.org/10.1007/978-3-319-75408-6_6

Aliaksei Severyn and Alessandro Moschitti. 2015. *UNITN: Training Deep Convolutional Neural Network for Twitter Sentiment Classification*. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*. Association for Computational Linguistics, pages 464-469. <https://doi.org/10.18653/v1/s15-2079>

Ayda-zade K., Rustamov S., Mustafayev E. 2013. *Sentiment Analysis: Hybrid Approach*. In *Transactions of Azerbaijan National Academy of Sciences. "Informatics and control problems"*. Volume XXXIII №6., p. 100-108.

Barkha Bansala and Sangeet Srivastava. 2018. *Sentiment classification of online consumer reviews using word vector representations*. *Procedia Computer Science*, 132:1147-1153. <https://doi.org/10.1016/j.procs.2018.05.029>

Bing Li, Keith C.C. Chan, Carol Ou and Sun Ruifeng. 2017. *Discovering public sentiment in social media for predicting stock movement of publicly listed companies*. *Information Systems*, 69(1):81-92. <https://doi.org/10.1016/j.is.2016.10.001>

B. K. Bhavitha, Anisha P. Rodrigues and Niranjana N. Chiplunkar. 2017. *Comparative study of machine learning techniques in sentiment analysis*. In *2017 International Conference on Inventive Communication and Computational Technologies (ICICCT)*. IEEE, pages 216-221. <https://doi.org/10.1109/ICICCT.2017.7975191>

D Shubham, P Mithil, Meesala Shobharani and S Sumathy. 2017. *Aspect level sentiment analysis using machine learning*. *IOP Conference Series: Materials Science and Engineering*, 263:042009. <https://doi.org/10.1088/1757-899x/263/4/042009>

Erik Cambria. 2016. *Affective Computing and Sentiment Analysis*. *IEEE Intelligent Systems*, 31(2):102-107. <https://doi.org/10.1109/MIS.2016.31>

Evangelos Psomakelis, Konstantinos Tserpes, Dimosthenis Anagnostopoulos and Theodora Varvarigou. 2014. *Comparing Methods for Twitter Sentiment Analysis*. In *Proceedings of the International Conference on Knowledge Discovery and Information Retrieval - Volume 1: KDIR*. SciTePress, pages 225-232. <https://doi.org/10.5220/0005075302250232>

Freha Mezzoudj and Abdelkader Benyettou. 2018. *An empirical study of statistical language models: N-gram language models vs. neural network language*

- models. *International Journal of Innovative Computing and Applications*, 9(4):189. <https://doi.org/10.1504/ijica.2018.10016827>
- Khanvilkar Gayatri and Vor, Deepali. 2018. Sentiment Analysis for Product Recommendation Using Random Forest. *International Journal of Engineering and Technology(UAE)*, 7(3):87-89. <https://doi.org/10.14419/ijet.v7i3.3.14492>
- Li I. Tan, Wai S. Phang, Kim O. Chin and Patricia Anthony. 2015. Rule-Based Sentiment Analysis for Financial News. In *IEEE International Conference on Systems, Man, and Cybernetics*. IEEE, pages 1601-1606. <https://doi.org/10.1109/SMC.2015.283>
- Mirsa Karim and Smija Das. 2018. Sentiment Analysis on Textual Reviews. *IOP Conference Series: Materials Science and Engineering*, 396:012020. <https://doi.org/10.1088/1757-899x/396/1/012020>
- Myint Zaw and Pichaya Tandayya. 2018. Multi-level Sentiment Information Extraction Using the CRbSA Algorithm. In *15th International Joint Conference on Computer Science and Software Engineering (JCSSE)*. IEEE, pages 1-6. <https://doi.org/10.1109/jcsse.2018.8457328>
- Nitika Nigam and Divakar Yadav. 2018. Lexicon-Based Approach to Sentiment Analysis of Tweets Using R Language. Singh M., Gupta P., Tyagi V., Flusser J., Ören T. (eds) *Advances in Computing and Data Sciences. ICACDS 2018. Communications in Computer and Information Science*, 905:154-164. https://doi.org/10.1007/978-981-13-1810-8_16
- Oscar Araque, Ignacio Corcuera-Platas, J. Fernando Sánchez-Rada and Carlos A. Iglesias. 2017. Enhancing deep learning sentiment analysis with ensemble techniques in social applications. *Expert Systems with Applications*, 77(1):236-246. <https://doi.org/10.1016/j.eswa.2017.02.002>
- Po-Hao Chen, Hanna Zafar, Maya Galperin-Aizenberg and Tessa Cook. 2017. Integrating Natural Language Processing and Machine Learning Algorithms to Categorize Oncologic Response in Radiology Reports. *Journal of Digital Imaging*, 31(2):178-184. <https://doi.org/10.1007/s10278-017-0027-x>
- Renata Lopes Rosa, Gisele Maria Schwartz, Wilson Vicente Ruggiero, Demóstenes Zegarra Rodríguez. 2019. A Knowledge-Based Recommendation System That Includes Sentiment Analysis and Deep Learning. *IEEE Transactions on Industrial Informatics*, 15(4):2124-2135. <https://doi.org/10.1109/tii.2018.2867174>
- Samir Rustamov. 2018. A Hybrid System for Subjectivity Analysis. *Advances in Fuzzy Systems*. Volume 2018: 9. <https://doi.org/10.1155/2018/2371621>
- Samir Rustamov, Elshan Mustafayev and Mark A. Clements. 2013. An Application of Hidden Markov Models in subjectivity analysis. In *2013 7th International Conference on Application of Information and Communication Technologies*. IEEE, pages 64-67. <https://doi.org/10.1109/ICAICT.2013.6722756>
- Samir Rustamov, Elshan Mustafayev and Mark A. Clements. 2013. Sentiment Analysis using Neuro-Fuzzy and Hidden Markov Models of Text. In *2013 Proceedings of IEEE Southeastcon*. IEEE, pages 1-6. <https://doi.org/10.1109/SECON.2013.6567382>
- Samir Rustamov and Mark Clements. 2013. Sentence-Level Subjectivity Detection Using Neuro-Fuzzy Model. In *Proceedings of the 4th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*. Association for Computational Linguistics, pages 108-114, Atlanta, USA. <https://www.aclweb.org/anthology/W13-1615>
- Sara Rosenthal, Noura Farra and Preslav Nakov. 2017. Sentiment analysis in Twitter. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*. Association for Computational Linguistics, pages 502-518. <https://doi.org/10.18653/v1/S17-2088>
- Umid Suleymanov and Samir Rustamov. 2018. Automated News Categorization using Machine Learning methods. *IOP Conference Series: Materials Science and Engineering*, 459:012006. <https://doi.org/10.1088/1757-899x/459/1/012006>
- Umid Suleymanov, Samir Rustamov, Murad Zulfugarov, Orkhan Orujov, Nadir Musayev and Azar Alizade. 2018. Empirical Study of Online News Classification Using Machine Learning Approaches. In *2018 IEEE 12th International Conference on Application of Information and Communication Technologies (AICT)*. IEEE, pages 1-6. <https://doi.org/10.1109/ICAICT.2018.8747012>
- Yashaswini Hegde and S.K. Padma. 2017. Sentiment Analysis Using Random Forest Ensemble for Mobile Product Reviews in Kannada. In *2017 IEEE 7th International Advance Computing Conference (IACC)*. IEEE, pages 777-782. <https://doi.org/10.1109/IACC.2017.0160>