ISSN No:-2456-2165

Opinion Mining in Albanian: Evaluation of the Performance of Machine Learning Algorithms for Opinions Classification

Nelda Kote
Faculty of Information Technology
Polytechnic University of Tirana
Tirana, Albania

Marenglen Biba
Faculty of Information Technology
New York University of Tirana, Albania
Tirana, Albania

Abstract:- The volume of opinions given in social media is growing, so we need useful tools to analyze them and use the gained information in the future. Opinion mining is an important research field to analyze the opinion given by someone for an entity. The most used methods in opinion mining are machine learning techniques. Our work aims to evaluate the performance of machine learning techniques through experiments in performing opinion classification tasks in the Al-banian language. Our approach to opinion mining addresses the problem of classifying text document opinions as positive or negative opinions. Since a lack of research on the Albanian language in this field, we conducted an experimental evaluation of fourteen techniques used for opinion mining. We tested different machine learning algorithm's performance using Weka. We have presented other conclusions related to the best feature combination of traditional machine learning algorithms.

Keywords:- *Opinion Mining, Sentiment Analysis, Machine Learning, Albanian Language.*

I. INTRODUCTION

Nowadays, social media has influenced the way people communicate with each other and express their opinions. People spend a considerable part of their day on social media, and opinions given about the products, services, or various social aspects posted on it are increasing day by day. The 2018 report of BrightLocal1 of "Local Consumer Review Survey" indicates that 86% of responders read online reviews, 50% of 18-34-year-old responders always read online reviews, and only 5% of them have never read online reviews. Opinion (or online review) analysis helps people in their decisions or companies to develop strategies to improve and expand their products or services. So, academia and business are interested in developing tools to analyze and extracts information to the huge number of online opinions.

Opinion mining aims to identify information about the sentiment expressed in an online opinion by a person. One of the tasks in opinion mining is opinion classification at the document level based on a pre-defined emotional expression category. For example, the category can contain two classes, the positive where are classified the opinions that express a positive opinion, and the negative where are classified the opinions that express a negative opinion. While extensive research work in opinion mining is concentrated on languages such as English, German, Italian or Japanese, insignificant work is done for the Albanian language.

The Albanian language is an independent branch of the Indo-European language family spoken by around 7 million native speakers. Albanian is the official language in Albania and Kosovo, the second official New Republic of Macedonia, and the regional language in the Ulcinj in Montenegro. Also, it is spoken in some provinces in southern Italy, Sicily, Greece, Romania, and Serbia, and by Albanian communities all over the world where Albanians have migrated and are living. Studying the Albanian language is interesting not only in linguistics but and in other fields since it is a unique and separate language. Albanian is a very complex language, its large alphabet contains 36 letters, and it has complex morphological and lexical features.

Here, we present an intensive study for opinion classification based on the opinion's polarity. The classification is done at the document level. The opinion is categorized as positive if it expresses a positive opinion and as negative if it expresses a negative one. Our work consists of evaluation through experiments the performance of machine learning approaches for the opinion classification task of opinion mining in in-domain and multi-domain corpora. The Albanian language is not a well-documented language in natural language processing resources, here are not available annotated corpora to be used for opinion mining tasks. For these reasons, we collected text documents opinions from online media to create a dataset with opinions classified into two classes, positive and negative. We have used different preprocessing tools for better performance. In the following section, we have discussed these in more detail.

In Section 2, we present a literature review of opinion mining. In section 3, we pre-sent a literature review of machine learning techniques. Section 4 describes the dataset collection, creation, and preprocessing. In section 5 are shown the experiment results for machine learning algorithms. And finally, the conclusion of our work is presented.

II. OPINION MINING

An opinion can be defined as: "a subjective statement, point of view, or emotion for an aspect of an entity or for the entity in general by the person who gives it". The opinion can be in one of the two forms: an opinion that gives an opinion about some-thing or a comparative opinion that gives an opinion about something comparing it with something else.

According to the definition in [1], an opinion is a quintuple, (ei, aij, sijkl, hk, tl), where ei is the name of an entity, aij is an aspect of ei, sijkl is the sentiment for aspect aij of entity ei, hk is the opinion holder, and tl is the time when the opinion is ex-pressed by hk. So, opinion mining must analyze and evaluate these five elements of the quintuple.

Opinion mining includes tasks as opinion classification based on a variety of pre-defined categories, subjectivity analyses of feelings, identification of opinion’s aspects and entities, summarization of opinions, identification of fake opinions, etc.

The techniques used in opinion mining can be machine learning techniques and lexicon-based techniques. In recent years, researchers more and more are combining these two types of techniques to have better results. Machine learning is an artificial intelligence field that studies techniques and algorithms that learn knowledge from annotated or not data and predict results for new data. Depending on the training and testing dataset used opinion mining can be in-domain, multi-domain, or cross-domain. In in-domain opinion mining, the train and test datasets are from the same domain. In multi-domain opinion mining, the train and test dataset contain opinions from multiple domains. In cross-domain opinion mining, the train and test dataset contain opinions from different domains.

In opinion mining, the opinion classification task can be in three levels of granularity: document, sentence, and aspect.

A. Document level opinion classification

The document level opinion classification can be the highest level of abstraction. At this level, an opinion document is considered as an entity, and to it is assigned an overall opinion polarity. In the two-level classification schema, positive and negative, the whole opinion document is considered to express a positive or a negative opinion. The aspects, entities, or sentiments described by the opinion are not analyzed in de-tail. Opinion classification at the document level can be a special case of traditional document classification.

B. Sentence level opinion classification

Sentence level opinion classification defines the polarity of the opinion expressed in a single sentence. In an opinion document (multi-sentence), each sentence is separately analyzed and assigned a polarity. The sentences in an opinion document can be subjective, that do not express an opinion, and objective, that express an opinion. In the case of determining the overall polarity of an opinion document, first, its sentences are classified as subjective or objective, and then to each objective sentence is assigned a polarity. The subjective sentences are disregarded. The overall polarity of the opinion document is estimated as the sum of the polarity of each sentence. In sentence level opinion classification is obtained more detailed information on opinion’s polarity than at the document level.

C. Aspect level opinion classification

Aspect level opinion classification considers all the aspects of the entity expressed in the opinion and determines the polarity of the opinion for each aspect. At document level, the classification of an opinion as positive or negative does not determine the polarity for each of the aspects of the entity. Aspect level classification aims to determine the polarity of an opinion document or sentence based on the opinion’s polarity of each aspect described.

For example, the sentence “The laptop memory is good, but the battery life is short." expresses a positive opinion for aspect “memory” and a negative opinion for aspect “battery”. First, you needed to identify the entities’ aspects, and then the polarity of the opinion expressed for each aspect.

In paper [1] the authors define that the aspect level considers the opinion itself and not its linguistic structure. The authors in the paper [2] conclude that this level of classification gives more accurate results.

III. MACHINE LEARNING TECHNIQUES

Machine learning techniques learn knowledge from annotated or not data and predict results for new data. These techniques are classified as supervised learning techniques, semi-supervised learning techniques, and unsupervised learning techniques. In our experimental evaluation, we focused on supervised learning techniques.

A. Supervised learning techniques

In opinion classification, the supervised machine learning algorithm aims to learn the sentiment polarity of opinions from an annotated dataset and predict the polarity of a new non-annotated opinion. One of the challenges in using these techniques is the need to have large, annotated datasets. This is time-consuming and requires good annotation skills to create the dataset. A sentiment can be expressed in different ways and a word can express a positive sentiment or negative sentiment depend on the subject the opinion is about. The performance of these techniques depends upon the domain and the number of opinions they are trained.

Naïve Bayes (NB), Support Vector Machine (SMV), K-NN, Maximum Entropy (ME), Decision Tree (DT) are the most popular machine learning algorithms used for opinion classification tasks in opinion mining.

The research work presented in [3] can be considered as one of the first attempts to use classification algorithms to classify opinions based on their sentiment polarity. They evaluate through experiments the performance of three supervised algorithms NB, SVM, and ME in a movie review dataset. The opinions are classified into two classes, as positive or negative opinions. The results indicate that these algorithms for opinion classification have the worst performance comparing to the topic-based classification. The SVM is the best performing algorithm, and Naïve Bayes is the worst performing algorithm.

In paper [4] the authors propose two methods to improve the performance of the classifier using fusing training data from multiple domains. They used the SVM algo-rithm to develop the classifier. The first method, feature-level fusion, combines the feature set from all domains into one feature set. In the second method, class-fusion level, the algorithm is separately trained for each domain and then the trained models are combined. The authors evaluated the performance of the proposed methods using a dataset with opinions from four different domains. Also, they evaluate the impact of n-grams on the performance of the classifier. The results indicate that the classifier-level method outperforms the feature-level method and single domain classification.

The paper [5] proposes a solution about how to address and avoid domain de-pendency and poor-quality annotated data. The proposed model learns high-level features from annotated and unannotated data that can be generalized across do-mains.

The authors in [6] proposed to use a mixed graph of terms, using a Latent Dirichlet Allocation (LDA) to grab the sentiments of opinions. The proposed method is evaluated to classify the opinion of the students extracted from e-learning platform.

Researchers are using artificial neural networks (ANN) for opinion mining more and more in recent years. The artificial neural networks are inspired by the structure of the human brain, consisting of a huge number of nodes called neurons. The neurons are information processing entities connected and organized in layers. The artificial neural network learns to fulfill a task in our case to classify the opinion based on their sentiment polarity by correcting the weights of connections between the nodes.

The experimental results presented in paper [7] demonstrate that the bag-of-words neural network model has better performance than NB and SMV algorithms for the opinion classification task. Also, in paper [8] the author demonstrates through the experimental results that the LSTM approaches in combination with Word2vec and GloVe embeddings are the best performing.

The research work mentioned above is mainly conducted for opinion mining in the English language. But a lot of research work is conducted in different languages like German, Greek, Italian, Chinese, Turkish, etc. In paper [9] is proposed a method to perform opinion classification in the German language using different features as lemmatization, part-of-speech tags, Named Entity Recognition, bag-of-words, and n-grams. The results on two different datasets show that the SVM classifier outperforms NB.

Greek is one of the most experimented languages for opinion mining. In [10], the SVM algorithm is used to implement an opinion classifier for hotel reviews using different features. The results indicate that the TF-IDF bag-of-words method is more powerful than the TO method.

The authors in the paper [11] evaluated the performance of the Rocchio algorithm, Naive Bayes classifier, and the combination of the two algorithms, to classify text opinions in the Italian language by their sentiment polarity. They used different pre-processing tools and features as part-of-speech tagging, single terms, n-grams, etc. The experimental results suggest the best settings to be used in this case to have good results.

The model proposed in [12] combines a supervised machine learning approach and a lexicon-based approach to perform the opinion classification task in the Turkish language. The authors developed a polarity lexicon named SentiTurkNet. The two used machine learning algorithms are NB and SVM combined with different features. They have almost the same performance.

Some research works present opinion classification tasks in the Albanian Language. The papers [13],[14], and [15] present the experimental performance evaluation of machine learning algorithms for opinion classification as positive and negative in in-domain and multi-domain corpora in the Albanian Language. By their results, we cannot define the best performing algorithms in terms of accuracy, but we can highlight a group of best-performing algorithms. The authors concluded that the opinions classification task depends on the domain and the number of opinions the algorithm is trained.

B. Semi-supervised learning techniques

The semi-supervised learning techniques use a small amount of annotated data and a large amount of unannotated data to build a model. Two of the most used semi-supervised learning techniques are co-training and self-training.

The self-training methods build a model by training a classifier using a small amount of annotated data, and then the model is used to label unannotated data. The most confident labeled data are then added to the first annotated. This larger corpus is then used to re-train the classifier and have a better model. This process is repeated one or more times depending on the amount of annotated data we want to have. One of the problems in these methods is if the first trained model mislabels the unannotated data. So, this mislabeled data will modify the model incorrectly. To address this, paper [16] proposed a self-training competitive method. The authors created three models by mixing three perceptive: the threshold, the same number, and the largest number of updates. The best model is chosen to get the highest F-measure.

The Co-training method separately trains two different classifiers for two different aspects of a small, tagged corpus. These models are used to label unlabeled test data. Each classifier is re-trained with the prediction results of the other classifier generating a large amount of labeled data.

The authors in [17] created a corpus of opinion tweets using two semi-supervised methods of self-training and co-training based on a relatively small corpus of labeled tweets. Experimental results show that the co-training method is best performing when we have limited labels whereas self-training is best performing when we have large amounts of labeled data.

C. Unsupervised learning techniques

Unsupervised learning techniques aim to define the hidden characteristics of un-annotated data. Clustering is one of the mainly used unsupervised techniques. Clustering is the process of dividing data into groups named clusters. The data within a cluster are very similar, whereas the data in different clusters are as diverse as possible.

In paper [18] the authors used a spectral clustering technique using the k-means algorithm to classify tweets as positive and negative. The experiment results show that this technique performs better than SVM, ME, and NB.

IV. CLASSIFIER SELECTION

In our previous works presented in papers [13] and [14], we evaluated the performance of 50 machine learning algorithms implemented in Weka for opinion classification in the Albanian language. The aim was to evaluate the performance of these algorithms to classify opinions in two classes: positive or negative. In [13], we evaluated the performance of these algorithms for in-domain opinion mining. To perform these experiments, we used 5 corpora, C_1 to C_5, specified in Table 1. Thus, in paper [14] we evaluated the performance of these algorithms for multi-domain opinion mining. To perform these experiments, we used 11 different corpora that vary from the number of the domains and the number of opinions used. One of the corpora is C_6 specified in Table 1. The results of in-domain opinion classification indicated that there are five best performing algorithms: Hyper Pipes, Logistic, Multi-Class Classifier, RBF Classifier, and RBF Network, and we could not define any statistical difference in performance between them. Even in the multi-domain opinion classification case, we could not define one best performing algorithm. The best performing algorithms are Naïve Bayes Multinomial Updateable, Complement Naïve Bayes, Naïve Bayes Multinomial, Logistic, SGD, Hyper Pipes, and RBF Network. The Logistic, Hyper Pipes, and RBF Network algorithms best perform in the two evaluations. We concluded that the algorithms' performance depends upon the number and the domain of opinions used in training, and in in-domain, the performance is better than in multi-domain.

To further investigate the features that impact the performance of the machine learning algorithms, we selected seven of the best-performing algorithms in our previous work, the Naïve Bayes Multinomial, Logistic, SGD, Hyper Pipes, and RBF Network. We left out the Multi-Class Classifier because it uses the Logistic algorithms and RBF Classifier. Also, we selected seven more algorithms from the list: the Bayesian Logistic Regression, SMO, Random Forest, Voted Perceptron, Simple Logistic, J48, and IBK due to their good results and popularity in sentiment analyzing re-searches. Table 2 shows the average value of the results for each of the algorithms published in papers [13] and [14].

V. THE DATASET CREATION

Taking into consideration the lack of Albanian linguistic resources as corpora, we decided to create our text opinions dataset.

A. Data Collection

To create the dataset, we collected 500 text documents written opinions in the Al-banian language from different Albanian newspapers. The opinions are related to 5 domains: higher education law, waste import, the impact of using in small businesses the VAT, tourism, and politics. Each of the collected opinions is manually clean and annotated based on the polarity of the sentiment they express in two classes: positive or negative sentiment. For each domain we collected 50 opinions annotated as positive and 50 opinions annotated as negative. Table 1 shows detailed information about created and used corpora.

Table 1 List of the created corpora

Corpus code	No. of domains	Domain Field	Total Opinions (Positive/Negative)
C_1	1	Tourism	100 (50/50)
C_2	1	Higher education law	100 (50/50)
C_3	1	Politics	100 (50/50)
C_4	1	VAT in Small Business	100 (50/50)
C_5	1	Waste import	100 (50/50)
C_6	5	Tourism, Higher education law, Politics, VAT in Small Business, Waste import	500 (250/ 250), (50 positive and 50 negative for each domain)

B. *Data Preprocessing*

Firstly, the text data are preprocessed with different tools, and then they are used to create a trained model for each classification algorithms and to evaluate the performance of the model. To evaluate the effectiveness of different preprocessing tools we decided to use the cases explained below.

First preprocessing case: firstly, on the text data is applied a stop-word removal that: converts the words to lower case and remove the stop-words, numbers, special characters, and punctuation marks. And secondly is applied a stemmer, the Albanian rule-based stemmer, to convert the words to their stem. The used stemmer is developed by [19] and [20] using java programming conforms to the morphological rules of the Albanian language. In it are implemented 134 morphological rules for generating the stem of a word but it does not take into consideration the word's linguistic meaning.

In the second preprocessing case we preprocessed the data using only the first step of the first case: the conversion of the words to lower case, the stop-words removal, and the special characters, numbers, and punctations removal.

By the end of this phase, the text documents of the opinions are bags of words without any language structures.

VI. EXPERIMENTAL EVALUATION

In this section, we discuss the performance of the selected machine learning algorithms to classify text document opinions based on their sentiment polarity in one of the two classes: positive and negative.

As mentioned in section 4, Table 2 are shown the experimental results of the selected algorithms from papers [13] and [14], in terms of the percent of correctly classified instances. We highlighted in black the best result for each corpus and italic the best result for each classifier. For each algorithm, we have calculated three average values: the results average for in-domain corpora, the results average for multi-domain corpora, and the average of all the results. To calculate the results average value of the multi-domain corpora, we have taken into consideration all the results presented in [14] that are not included in the table.

Analyzing the results, we can conclude that the algorithms best perform when they are used to classify opinions in in-domain opinion mining. The best performing algorithm in terms of the average value is the Naïve Bayes, and its performance is more stable than the other algorithms.

We decided to investigate more features that can improve the performance of the algorithms listed in Table 2. To evaluate the performance of the selected algorithm we have to follow the steps described below.

Table 2 Experimental results from paper [13] and [14] in terms of percent of correctly classified instances

	Average In-domain	Average Multi-domain	Average Total
Naïve Bayes Multinomial	**83.40**	**79.35**	**80.62**
Logistic	81.00	77.61	78.67
Bayesian Logistic Regretion	80.60	77.38	78.39
SGD	80.60	76.13	77.53
SMO	79.40	74.70	76.17
Random Forest	75.40	75.03	75.15
HyperPipes	83.20	70.11	74.20
VotedPerceptron	76.40	70.57	72.39
SimpleLogistic	71.60	65.32	67.29
RBFNetwork	75.60	62.19	66.38
J48	66.80	63.43	64.49
IBK	61.80	61.04	61.28

The first step is the preprocessing of the text opinions. In the above-mentioned experiments, the text opinions are preprocessed using the first case of preprocessing detailed in section 5.2 (stop words, numbers, punctuation, and special character removal and stemming). In the following experiments, we decided to use the second case of preprocessing that includes only the stop words, numbers, punctuation, and special character removal. The words within an opinion document are not stemmed. At the end of this step, the text document can be considered as a bag-of-words.

The second step is the creation of an ARFF file in Weka per each corpus. To create the file per each corpus, we loaded all opinion documents in Weka using textDirectoryLoader class. Next, the StringToWordVector filter is applied with different features selected for converting the string attributes into a word vector. We used six different feature selection in StringToWordVector filter:

a) *WordTokenizer feature*
b) *WordTokenizer feature and TF-IDF*
c) *n-gram configuration min=1 and max=2*
d) *TF-IDF and n-gram configuration min=1 and max=2*
e) *n-gram configuration min=1 and max=3*
f) *TF-IDF and n-gram configuration min=1 and max=3*

In the experiments presented in paper [13] and [14], the WordTokenizer feature is used.

The third step is to train and test the model. The ARFF files are used in Weka Explores, with 10 folds cross-validation feature selected, to train a model per each algorithm. The performance of each model is evaluated in terms of the percent of correctly classified instances.

The results of the experiments are shown in Table 3. Column a_1 presents the experimental values from our previous work in [13] and [14]. The other columns present the results with one of the feature combinations. The best result per corpus is highlighted in bold. In three of the corpora, the best performant algorithm is RBF Network. Hyper Pipes is the best per-formant algorithm in two corpora and Naïve Bayes Multinomial only in one. We calculated the average value of all the results per algorithm. The best performing algorithm is Naïve Bayes Multinomial, with 84.88% of correctly classified instances. The performance of the algorithms is improved when the stemmer is not used. The algorithms perform better when 2-grams with or without TF-IDF are used. Further-more, for each algorithm, we calculated the average value of the results per feature used. Even in this case, Naïve Bayes Multinomial is the best performing algorithm. These algorithms best perform when feature d is used. The Naïve Bayes Multinomial algorithm best performs when TF-IDF and n-gram with values min=1 and max=2 is used.

To analyses, if there is any statistical performance difference between the algorithms, we perform another experiment using the Weka Experimenter tool. Based on the average value of percent correctly classified instances from Table 3, we choose as the base algorithm Naïve Bayes Multinomial. The experiment is performed using all the algorithms, ten repetitions, and ten cross-validations. Table 4 are shown the results in terms of the percent correct classified. The * annotation in the results indicates that the result is statistically worse than the baseline scheme, Naïve Bayes Multinomial. In the results, there is not a v annotation so we can define that none of the algorithms perform statistically better than the baseline scheme Naïve Bayes. In the last row of the table per each algorithm is a count value of the form (a/b/c) indicating the number of times the algorithm has been (a) better, (b) the same, and (c) worse than the base algorithm.

The Simple Logistic, J48, IBk algorithm have been worse than Naïve Bayes all seven times. The RBF Network and Bayesian Logistic Regression algorithms have been the same as Naïve Bayes for six-time and only and only one time perform worse than it. So, we cannot define any statistical difference between these three algorithms.

Table 5 shows the summary test of the above experiment. The number out of the brackets is the times that the algorithm in the column is better than it in the row, a 0 means that the algorithm corresponding to the column did not get a win versus the algorithm corresponding to the row. The number in brackets is the number of significant wins of the algorithm corresponding to the column versus the algorithm corresponding to the row.

To rank the algorithm based on the result of the experiment, we perform the ranking test. The ranking result is showed in Table 6. This test ranks each algorithm based on the total number of wins and losses they archive versus all the other algorithms. In column > is the number of wins, in column < is the number of losses, and in column > − < is the difference between the number of wins and losses per each algorithm. The Naïve Bayes algorithm is ranked in the first place and RBF Network algorithm in the second place. The other algorithms have a small number or even negative for the five last algorithms listed in the table. This indicates that they do not have a good performance comparing to the other algorithms.

Table 3 Experimental results in terms of percent of correctly classified instances

	C_1							C_2							C_3						
Feature	***a_1***	***a***	***b***	***c***	***d***	***e***	***f***	***a_1***	***a***	***b***	***c***	***d***	***e***	***f***	***a_1***	***a***	***b***	***c***	***d***	***e***	***f***
NaiveBayesMultinomial	87	92	92	93	**93**	93	93	89	95	95	94	97	94	95	77	82	82	84	87	86	89
SGD	88	92	92	92	92	93	93	84	89	89	89	89	87	87	75	83	83	82	82	83	83
RBFNetwork	49	93	93	51	51	95	51	88	50	92	97	97	97	97	66	90	90	**92**	**92**	86	86
BayesianLogisticRegretion	89	91	90	89	89	89	89	85	92	91	90	94	91	93	77	81	82	86	85	87	85
Logistic	*94*	92	92	93	**93**	93	93	84	94	94	84	84	87	87	66	82	82	90	90	82	82
HyperPipes	92	92	92	91	91	92	92	*92*	93	93	**98**	**98**	**98**	**98**	67	80	80	88	88	82	82
SMO	88	90	90	92	92	91	91	82	87	87	82	82	81	81	74	82	82	81	81	84	84
Random Forest	86	83	81	88	90	90	88	76	80	83	67	78	79	77	*76*	79	77	78	79	79	82
VotedPerceptron	86	82	82	81	82	80	82	76	88	84	87	88	88	83	74	79	77	80	78	82	80
SimpleLogistic	84	80	80	82	81	80	82	74	68	65	70	68	67	65	66	69	68	71	68	68	68
J48	87	72	72	70	70	70	70	67	57	57	57	57	59	59	58	76	76	65	65	65	65
IBK	60	63	63	63	63	67	67	65	64	64	65	65	65	65	62	60	60	56	56	62	62

(continued)

Table 3 (continued)

	C_4							C_5							C_6						
Feature	***a_1***	***a***	***b***	***c***	***d***	***e***	***f***	***a_1***	***a***	***b***	***c***	***d***	***e***	***f***	***a_1***	***a***	***b***	***c***	***d***	***e***	***f***
NaiveBayesMultinomial	79	83	83	87	89	87	87	85	86	84	87	90	89	89	*78.6*	77	78	78	**79.4**	78	79
SGD	75	76	76	76	76	71	71	81	80	80	87	87	84	84	77.4	75	75	75	75.2	76	76
RBFNetwork	*86*	88	88	90	90	89	89	*89*	51	93	**93**	**93**	**93**	**93**	73.4	76	76	76	76.2	74	74
BayesianLogisticRegretion	74	72	72	70	72	71	71	78	76	80	78	82	77	81	77	76	76	76	77.2	75	76
Logistic	82	86	86	88	88	85	85	79	80	80	86	86	89	89	66.6	68	68	67	67.4	64	64
HyperPipes	85	88	88	**92**	**92**	**92**	**92**	80	81	81	90	90	90	90	55.8	59	59	62	62.2	60	60
SMO	75	71	71	73	73	74	74	78	76	76	79	79	81	81	76	72	72	72	71.6	71	71
Random Forest	65	74	67	70	68	67	70	74	74	78	77	76	82	75	74.2	77	60	77	76.8	77	78
VotedPerceptron	71	70	76	68	74	72	75	75	76	73	78	82	77	79	69.2	70	70	72	71.6	71	73
SimpleLogistic	63	67	67	69	71	69	72	71	61	61	63	63	66	66	66.4	69	69	68	68.2	67	67
J48	60	53	53	58	58	58	58	62	67	67	64	64	75	75	61.2	64	64	66	65.8	66	66
IBK	60	62	62	68	68	65	65	62	50	50	58	58	59	59	60.4	59	59	60	60.4	60	60

(continued)

Table 3 (continued)

	C_7							**Average per each feature**							**Avg. Total**
Feature	***a_1***	***a***	***b***	***c***	***d***	***e***	***f***	***a_1***	***a***	***b***	***c***	***d***	***e***	***f***	
NaiveBayesMultinomial	74.11	74.22	75.8	72.9	73.9	72.9	73.8	*81.39*	*84.15*	84.31	*85.07*	**87.04**	85.7	*86.51*	**84.88**
SGD	72.44	72.89	72.9	74.6	74.6	74.2	74.2	78.98	81.13	81.13	82.25	82.25	81.12	81.12	81.14
RBFNetwork	70.44	71.89	71.9	69.2	69.2	69.8	69.8	74.55	74.3	*86.3*	81.2	81.2	*86.31*	80.03	80.56
BayesianLogisticRegretion	73.89	73.11	73.9	72.2	73.1	72.7	73.1	79.13	80.1	80.7	80.12	81.76	80.35	81.13	80.47
Logistic	59.44	61.33	61.3	59.6	59.6	60.4	60.4	75.86	80.53	80.53	81.14	81.14	80.12	80.12	79.92
HyperPipes	50.67	49.67	49.7	49.7	49.7	49.7	49.7	74.64	77.47	77.47	81.55	81.55	80.52	80.52	79.1
SMO	*71.67*	72.33	72.3	72	72	71.8	71.8	77.81	78.56	78.56	78.66	78.66	79.08	79.08	78.63
Random Forest	75.56	**77.1**	75.4	74.7	75	75.1	75.6	75.25	77.73	74.43	76.01	77.54	78.5	77.91	76.77
VotedPerceptron	70.89	71.56	73.3	70.6	70	70.2	68.8	74.58	76.68	76.48	76.68	77.94	77.15	77.25	76.68
SimpleLogistic	68.22	69.78	69.8	68.4	68.4	69.1	69.1	70.37	69.08	68.51	70.23	69.66	69.44	69.87	69.6
J48	60	62.11	62.1	61.3	61.3	61.3	61.3	65.03	64.42	64.42	63.02	63.02	64.88	64.88	64.24
IBK	60.67	58.11	58.1	57.8	57.8	58.1	58.1	61.44	59.47	59.47	61.17	61.17	62.3	62.3	61.05

Table 4 Statistical Experiment results per each algorithm in terms of percent correct

Dataset	**Naïve Bayes Multinomial**	**SGD**	**Bayesian Logistic Regression**	**Hyper Pipes**	**RBF Network**	**Logistic**	**SMO**	**Voted Perceptron**	**Random Forest**	**Simple Logistic**	**J48**	**IBk**
C_1	92.10 |	91.4	87.6	91.5	95.1	92.5	89.7	83.60*	85.3	77.60*	74.30*	63.20*
C_2	96.20 |	88.30*	91.7	97.9	97.4	90.5	83.60*	86.70*	80.70*	66.60*	58.80*	63.40*
C_3	85.30 |	83.1	86.9	87.4	90.9	84.5	80.5	78.3	79.4	68.90*	67.50*	57.40*
C_4	88.80 |	76.50*	76.70*	93	93.3	88.2	75.00*	72.90*	71.50*	65.30*	58.90*	66.60*
C_5	87.80 |	81.1	82.5	90	93.1	86.2	80.2	76.40*	78.8	66.10*	65.40*	57.80*
C_6	79.06 |	74.76*	76.36	61.72*	76.34	67.84*	71.44*	73.02*	77.68	67.40*	64.28*	60.12*
C_7	73.72 |	73.44	72.33	49.87*	68.86*	59.11*	71.28	71.44	75.19	67.60*	61.21*	58.72*
	(v/ /*) |	(0/4/3)	(0/6/1)	(0/5/2)	(0/6/1)	(0/5/2)	(0/4/3)	(0/2/5)	(0/5/2)	(0/0/7)	(0/0/7)	(0/0/7)

Table 5 Summary test results of the experiment for each algorithm

a	b	c	d	e	f	g	h	i	j	k	l	|(No. of datasets where [col] >> [row])
-	0 (0)	1 (0)	4 (0)	5 (0)	1 (0)	0 (0)	0 (0)	1 (0)	0 (0)	0 (0)	0 (0)	|a = (1) NaiveBayesMultinomial
7 (3)	-	5 (0)	5 (3)	6 (3)	5 (1)	0 (0)	0 (0)	2 (0)	0 (0)	0 (0)	0 (0)	|b = (2) SGD
6 (1)	2 (0)	-	5 (1)	5 (3)	3 (1)	1 (0)	0 (0)	2 (1)	0 (0)	0 (0)	0 (0)	|c = (3) BayesianLogisticRegression
3 (2)	2 (2)	2 (2)	-	6 (2)	3 (2)	2 (2)	2 (2)	2 (2)	2 (2)	2 (1)	1 (1)	|d = (4) HyperPipes
2 (1)	1 (1)	2 (0)	1 (0)	-	0 (0)	1 (0)	1 (0)	2 (1)	0 (0)	0 (0)	0 (0)	|e = (5) RBFNetwork
6 (2)	2 (2)	4 (2)	4 (0)	7 (2)	-	2 (1)	2 (1)	2 (2)	1 (1)	1 (0)	0 (0)	|f = (6) Logistic
7 (3)	7 (2)	6 (2)	5 (3)	6 (3)	5 (1)	-	3 (0)	2 (2)	0 (0)	0 (0)	0 (0)	|g = (7) SMO
7 (5)	7 (1)	7 (0)	5 (4)	6 (5)	5 (2)	4 (0)	-	5 (0)	0 (0)	0 (0)	0 (0)	|h = (8) VotedPerceptron
6 (2)	5 (1)	5 (1)	5 (3)	5 (5)	5 (2)	5 (0)	2 (0)	-	0 (0)	0 (0)	0 (0)	|i = (9) RandomForest
7 (7)	7 (7)	7 (6)	5 (5)	7 (6)	6 (5)	7 (4)	7 (2)	7 (5)	-	0 (0)	1 (0)	|j = (10) SimpleLogistic
7 (7)	7 (7)	7 (7)	5 (5)	7 (7)	6 (5)	7 (7)	7 (4)	7 (4)	7 (1)	-	2 (0)	|k = (11) J48
7 (7)	7 (6)	7 (7)	6 (5)	7 (7)	7 (6)	7 (6)	7 (6)	7 (6)	6 (4)	5 (0)	-	|l = (12) IBk

Table 6 Ranking test results for each algorithm

Resultset	>-<	>	<
NaiveBayesMultinomial	40	40	0
RBFNetwork	38	43	5
BayesianLogisticRegression	18	27	9
SGD	13	29	16
Logistic	8	25	17
RandomForest	5	23	18
HyperPipes	5	29	24
SMO	-2	20	22
VotedPerceptron	-12	15	27
SimpleLogistic	-53	8	61
J48	-67	1	68
IBk	-73	1	74

VII. CONCLUSIONS

This paper presents a performance evaluation of machine learning algorithms for opinion mining in the Albanian language. Our approach, in opinion mining, is to classify an opinion based on the sentiment polarity it expresses. The opinion is classified into one of the two categories, positive or negative. If the sentiment polarity of an opinion is positive, the opinion is categorized as positive. And, if the sentiment polarity of an opinion is negative, the opinion is categorized as negative. Through experiments, we have evaluated the performance of twelve classification algorithms implemented in the Weka program. To perform the evaluation, we have used different corpora collected by ourselves. Five corpora contain opinions from only one domain, and one corpus contains opinions from multiple domains. Before use to train and test a model, the text documents of the corpora are passed in a preprocessing phase. The preprocessing phase includes the conversion of the words to lower case, the stop-words removal, and the special characters, numbers, and punctations removal. In the performed experiments to evaluate the performance of the algorithms implemented in Weka, we have used different features in the StringToWordVector filter as WordTokenizer, TF-IDF, and n-gram. Comparing the experimental results with the results of papers [13] and [14], we can conclude that the performance of the algorithms is better when the stemmer is not used. The performance of the algorithms is improved when TF-IDF and n-gram (with values min=1 and max=2) are used. Naïve Bayes Multinomial and RBF Network are the best performing algorithms.

REFERENCES

[1]. B. Liu, Sentiment Analysis: mining sentiments, opinions, and emotions, 2nd ed. Cambridge University Press. ISBN 9781139084789, 2015.

[2]. Ch. SG Khoo and S.B. Johnkhan, “Lexicon-based sentiment analysis: Comparative evaluation of six sentiment lexicons,” Journal of Information Science, vol. 44, no. 4, pp. 491-511, 2018.

[3]. B. Pang, L. Lee and S. Vaithyanathan, “Thumbs up?: sentiment classification using machine learning techniques,” In Proc. Conference on Empirical Methods in Natural Language Processing ‘02, 2002, pp. 79–86.

[4]. Shoushan L. and C. Zong, “Multi-domain Sentiment Classification,” In Proc. ACL-08: HLT ‘08, 2008, pp. 257–260.

[5]. F. Yang, A. Mukherjee and Y. Zhang, “Leveraging Multiple Domains for Sentiment Classification,” In Proc the 26th International Conference on Computational Linguistics’16, 2016, pp. 2978–2988.

[6]. F. Colace, M. D. Santo and L. Greco, “SAFE: A Sentiment Analysis Framework for E-Learning,” International Journal of Emerging Technologies in Learning, vol. 9, no. 6, pp. 37-41, 2014.

[7]. R. Moraes, J. F. Valiati and W. P. G. Neto, “Document-level sentiment classification: an empirical comparison between SVM and ANN,” Expert Systems with Applications, vol. 40, no. 2, pp. 621-633, 2013.

[8]. J. Barry, “Sentiment Analysis of Online Reviews Using Bag-of-Words and LSTM Ap-proaches,” In Proc. The 25th Irish Conference on Artificial Intelligence and Cognitive Science ‘17, 2017, pp. 272-274.

[9]. D. Ploch, “Intelligent News Aggregator for German with Sentiment Analysis,” Smart Information Systems. Advances in Computer Vision and Pattern Recognition. Springer, Cham, pp 5-46, 2015.

[10]. G. Markopoulos, G. Mikros, A.Iliadi and M. Liontos, “Sentiment Analysis of Hotel Reviews in Greek: A Comparison of Unigram Features,” Cultural Tourism in a Digital Era. Springer Proceedings in Business and Economics. Springer, Cham, pp. 373-383, 2015

[11]. S. Ferilli, B. De Carolis, F. Esposito and D. Redavid, “Sentiment Analysis as a Text Categorization Task: A Study on Feature and Algorithm Selection for Italian Language,” In Proc. IEEE International Conference on Data Science and Advanced Analytics ’15, 2015, pp. 1-10.

[12]. G. Gezici and B. Yanıkoglu, “Sentiment Analysis in Turkish,” Turkish Natural Language, Theory and Applications of Natural Language Processing, pp. 255-271, 2018

[13]. N. Kote, M. Biba and E. Trandafili, “A Thorough Experimental Evaluation of Algorithms for Opinion Mining in Albanian,” In Proc International Conference on Emerging Internetworking, Data & Web Technologies ’18, 2018, pp. 525-536.

[14]. N. Kote, M. Biba and E. Trandafili, “An Experimental Evaluation of Algorithms for Opinion Mining in Multi-Domain Corpus in Albanian,” In Proc. International Symposium on Methodologies for Intelligent Systems ‘18, 2018, pp. 439-447.

[15]. M. Biba and M. Mane, “Sentiment Analysis through Machine Learning: An Experimental Evaluation for Albanian,” Recent Advances in Intelligent Informatics, Advances in Intelligent Systems and Computing, vol. 235, pp. 195-203, 2014.

[16]. S. Hong, J. Lee and J. H. Lee, “Competitive self-training technique for sentiment analysis in mass social media,” In Proc, Joint 7th International Conference on Soft Computing and Intelligent Systems (SCIS) and 15th International Symposium on Advanced Intelligent Systems (ISIS) ‘14, 20114, pp. 9-12.

[17]. V. V. Asch and W. Daelemans, “Predicting the Effectiveness of Self-Training: Application to Sentiment Classification,” ArXiv, 2016.

[18]. V. Iosifidis and E. Ntutsi, “Large scale sentiment learning with limited label,” In Proc. 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining ’17, 2017, pp. 1823–1832.

[19]. J. Sadiku and M. Biba, “Automatic Stemming of Albanian Through a Rule-based Approach,” Journal of International, Research Publications: Language, Individuals and Society, vol. 6, pp. 173-190, 2012.

[20]. Biba M. Gjati E.: Boosting text classification through stemming of composite words. ISI, 185194, 2013.