American Journal of Engineering Research (AJER)
e-ISSN: 2320-0847 p-ISSN : 2320-0936
Volume-12, Issue-10, pp-91-97
www.ajer.org
Research Paper Open Access

A new approach, by combination of SVM with attention mechanism and bi LSTM for sentiment analysis of Bulgarian user comments

Petrova Daniela, Bozhikova Violeta
[1](Technical University, Varna, Bulgaria
Corresponding Author: Petrova D.

ABSTRACT : *Sentiment analysis, as part of Natural Language Processing, is getting more and more interest with the growing amount of web documents. This provoked the research of the authors in this field regarding the Bulgarian language. The purpose of the current paper is to propose a new ensemble method for sentiment analysis of user comments in Bulgarian language, called att_SVM+biLSTM+lex_RF. It combines two machines learning methods – Support Vectors Machines (SVM) with attention mechanism and Recurrent Neural Network (RNN) in a stacking ensemble method, and applies Random Forest (RF) as meta classifier with a partial lexicon based approach. The proposed method gives higher accuracy than the previously experimental studies of the authors.*
KEYWORDS *sentiment analysis, Bulgarian, user comments, ensemble methods.*

Date of Submission: 15-10-2023 Date of acceptance: 30-10-2023

I. INTRODUCTION

Nowadays, when everyday life is so connected with technology and Internet, the growing amount of web documents is vast. This leads to the urgent need of techniques and methods to make these web documents understandable to the computers. This task is performed and solved by Text Mining and all its subdivisions, like Natural Language Processing, Data Mining, and Machine Learning. Sentiment analysis, in turn, is part of Natural Language Processing and deals with the problem of categorizing user reviews and comments as positive or negative by extracting the opinion and the sentiment of every sentence. There are numerous instruments, techniques and applications, developed to extract sentiment of such texts regarding the English language. For not so widely used languages, like Bulgarian, the situation is different. There are only a few ready to use instruments for preprocessing, for example, almost none databases and the research made in this field is young and there is a lot to improve.

All of the above sparked the interest of the authors to start a research in the field of sentiment analysis for Bulgarian language. As a first step they have created two large databases with around 100 00 user comments each in Bulgarian, as well as their own stop words list to be used in the preprocessing part. Next they have experimented with the most common machine learning algorithms to find the most suitable for sentiment analysis of Bulgarian texts. Following this line of work in search for an algorithm that gives the highest results, in the current paper the authors present an improved approach, called **att_SVM+biLSTM+lex_RF,** by combining two machines learning methods– Support Vectors Machines (SVM) and Recurrent Neural Network (RNN) in a stacking ensemble method, and applying Random Forest (RF) as meta classifier. To further improve the prediction accuracy before applying SVM is implemented an attention mechanism and additionally is used a lexicon based approach in the RF classifier.

II. METHODS

Ensemble learning is a broad meta-approach in machine learning, aiming to improve predictive performance by amalgamating forecasts from multiple models. While there are an almost endless variety of ensembles that can be created for predictive modeling, three dominant methods stand out in the field of

ensemble learning. These methods have become entire domains of study, rather than mere algorithms. The primary categories of ensemble learning techniques encompass bagging, stacking, and boosting. [1]

- Bagging, which entails fitting numerous decision trees on different subsets of the same dataset and then averaging their predictions.
- Stacking, which involves fitting various types of models to the same data and using another model to determine the optimal way to combine these predictions.
- Boosting, a technique that sequentially introduces ensemble members to correct the predictions of prior models, ultimately yielding a weighted average of the predictions.

The chosen ensemble method for this project - Stacked Generalization, commonly known as stacking, is an ensemble technique designed to create a diverse group of members by employing various types of models on the training data and then utilizing another model to amalgamate their predictions. In the realm of stacking, there's a distinct terminology. Ensemble members are often referred to as "level-0 models" and the model responsible for combining predictions is known as the "level-1 model". The standard structure involves a two-level hierarchy of models, although more layers can be incorporated if needed. For instance, instead of a solitary level-1 model, one might opt for three or five level-1 models, with a single level-2 model tasked with aggregating the predictions from the level-1 models to make a final prediction. Any machine learning model can be employed for aggregating the predictions, although linear models like linear regression for regression tasks or logistic regression for binary classification are commonly used. This approach encourages the lower-level ensemble member models to encompass complexity, while simpler models are employed to learn how to effectively leverage the diversity of predictions.[1]

The key elements of stacking can be summarized as follows:

- Utilization of an unchanged training dataset;
- Adoption of different machine learning algorithms for each ensemble member;
- Deployment of a machine learning model to learn the optimal way to combine predictions;
- Diversity is achieved by employing various machine learning models as ensemble members.

Hence, the goal is to employ a diverse set of models that are either learned or constructed in distinct ways, ensuring that they make different assumptions and, as a result, exhibit less correlated prediction errors.

The other approach used in the proposed method is the use of attention mechanism. Attention mechanisms are a crucial component in many deep learning models, particularly in natural language processing (NLP) and computer vision tasks. They allow the model to focus on specific parts of the input data when making predictions or generating output. The concept of attention is inspired by human visual attention, where people selectively focus on certain regions of an image or specific words in a sentence while comprehending the overall context. In the context of sentiment analysis, attention mechanisms can be used to enhance the model's ability to understand and interpret the sentiment expressed in a piece of text, such as a sentence or a document. Attention mechanisms are particularly useful when dealing with long or complex texts, as they allow the model to focus on the most relevant words or phrases while making sentiment predictions.

In summary, attention mechanisms in sentiment analysis help models focus on the most informative parts of the text while making sentiment predictions. They enhance the model's performance, especially when dealing with lengthy or context-rich texts, and provide insights into the reasoning behind sentiment predictions.

Attention-based techniques have found extensive use in a variety of natural language processing tasks, including sentiment analysis, opinion mining, and recognizing emotions in text. Bahdanau and colleagues [3] introduced an encoder-decoder method with an integrated attention mechanism in the decoder. This mechanism repeatedly accesses the representation of a source sentence generated by the encoder, allowing the model to focus on relevant parts when predicting a target word. As a result, this approach achieved very high results.

Munkhdalai and team [4] proposed the "neural tree indexer" (NTI), which makes a balance between recurrent and recursive neural networks. NTI is a tree-structured model independent of syntactic parsing. It learns representations for premises and hypotheses, combining them using an attention mechanism to achieve high accuracy in classification tasks.

Yang and colleagues [5] introduced the "hierarchical attention network" (HAN) for document classification. HAN mirrors the hierarchical structure of documents and applies two levels of attention mechanisms at word and sentence levels. This enables the model to differentiate attention to more and less informative content while considering context, outperforming previous methods on six datasets.

Yin and team [6] extended the HAN model to propose a "hierarchical iterative attention model" for aspect-based sentiment analysis. This task is formulated as a machine comprehension problem, and the model surpasses the performance of the hierarchical attention network baseline.

Lee and colleagues [7] presented a method for identifying keywords that discriminate positive and negative sentences, using weakly supervised learning based on a convolutional neural network (CNN). The CNN model is trained on sentence matrices, and a word attention mechanism identifies high-contributing words

using a class activation map (CAM) generated from the CNN's fully connected layer. This approach provides sentence-level and word-level polarity scores using only weak labels.

Lin and team [8] proposed a model for sentence embedding using self-attention. This technique represents each sentence as a 2-D matrix, with each row attending to different parts of the sentence. It achieves high accuracy in both sentiment classification and textual entailment tasks.

Chen and colleagues [9] introduced an LSTM model that incorporates global user preference and product characteristics in sentiment classification. This hierarchical LSTM generates sentence and document representations and uses an attention mechanism on user and product information, considering information at both word and semantic levels.

Wang and colleagues [10] presented an attention-based LSTM with target embedding for aspect-level sentiment classification. This mechanism ensures that the model attends to the relevant parts of a sentence when different aspects are provided as input, achieving state-of-the-art performance.

Liu and team [11] proposed an AC-BiLSTM architecture, combining attention-based bidirectional LSTM with a convolutional layer for text classification. This model outperforms other accurate methods in sentiment analysis and highlights the effectiveness of BiLSTM over the convolution layer.

In addition to ensemble methods and an attention mechanism, in the proposed method is partially included a lexicon-based approach. A lexicon-based approach in sentiment analysis, also known as a dictionary-based approach, relies on predefined lexicons or sentiment dictionaries to assess the sentiment of text data. The fundamental idea behind this approach is to use a collection of words or phrases that are associated with specific sentiment polarities (positive, negative, or neutral) and then analyze the sentiment of a given text based on the presence and frequency of these sentiment-indicative terms. Lexicon-based sentiment analysis is relatively simple and interpretable, making it a good choice for certain applications. However, it has limitations in handling context, sarcasm, and nuances in language. For more advanced sentiment analysis tasks, machine learning-based approaches, such as supervised classification models or neural networks, are often preferred as they can capture context and learn from data. Lexicon-based methods can still be valuable in combination with other techniques to improve sentiment analysis accuracy.

III. Results and Discussion

As said above the others have created to databases with user reviews in Bulgarian with almost 100 000 comments each. [12] After the preprocessing of the data, the two databases have been equalized in order to be able correctly to compare and analyze the results. The table I is shown the total count of the reviews in each databases as well as the count of the positive and negative user comments. It is obvious that the negative comments are a lot less than the positive, which later led to the application of semi lexicon-based approach, using negative words only to boost the correct prediction of negative comments.

Table I Number of reviews in Database 1 and Database 2

User reviews	*Database 1*	*Database 2*
Positive	63 714	63 714
Negative	14 357	14 357
Total	**78 071**	**78 071**

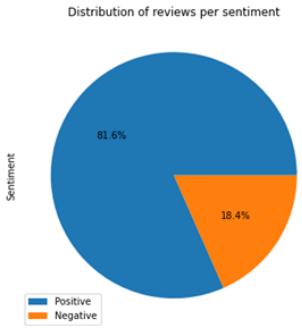

Figure 1 – Distribution of reviews per sentiment

In their previous works and publications, the authors have explored the different most common machine learning methods on the two self-created databases in Bulgarian language.[12,13] They have given quite satisfactory results, but such that could be improved. As a next step, was used an approach that combines two machine learning methods, executed through an ensemble method called Stacking. As stated above, stacking involves training multiple machine learning models and combining their predictions to improve overall accuracy. In this case, four combinations of methods were considered:

- SVM and RF;
- SVM and LSTM;
- SVM and LSTM with an attention mechanism;
- SVM with an attention mechanism and LSTM.

These combinations were chosen because they were the models that have achieved the highest accuracy in previous research studies. The parameters, chosen for SVM, RF, and LSTM are the once that yielded the best results in the previous computations that have been made. For all combinations was experimenting on two different splits between training and testing data 80%-20% and 70%-30% to choose which is more accurate for the two databases. In addition, both unigrams and bigrams in the vectorizing process were considered in the analysis, using TF-IDF vectorization.

In the first approach, separate models are trained, including SVM (with a LinearSVC function and parameters kernel=hinge and C=1) and a Random Forest classifier. A meta-classifier, specifically a logistic regression model, is then applied to the combined predictions.

Table II SVM + Random Forest results

SVM - Random forest				
Train-Test 70%-30%	*Bi-grams*		*Uni-grams*	
	Stemmed	*Not stemmed*	*Stemmed*	*Not stemmed*
Database 1	**0,902**	**0,903**	0,892	0,899
Database 2	0,942	**0,943**	0,931	0,939
Train-Test 80%-20%	*Bi-grams*		*Uni-grams*	
	Stemmed	*Not stemmed*	*Stemmed*	*Not stemmed*
Database 1	**0,903**	**0,904**	0,892	**0,903**
Database 2	0,942	**0,943**	0,933	0,939

It is evident from Table II that the difference between the two types of data splits for training and testing is negligibly small. There is also a slight difference between the stemmed and non-stemmed data, favoring the use of unprocessed data. However, the highest results are generated by the combination of not stemming and bigrams, achieving 90.3% for Base 1 and 94.3% for Base 2, respectively.

In the second combination of models, SVM with a LinearSVC function and parameters kernel=hinge and C=1 is used in conjunction with a two-layer LSTM (Long Short-Term Memory) model with the following layers:

- Embedding Layer: This layer converts input integers into dense vectors with a fixed size (in this case, 32) and represents each word in the input sequence.
- Bidirectional LSTM Layer (First Layer): This layer processes the input sequence both forwards and backwards, returning sequences for each time step.
- Bidirectional LSTM Layer (Second Layer): Similar to the first layer, this one also processes sequences bidirectionally but does not return sequences for each time step. Instead, it provides a summarized representation of the entire sequence.
- Dense Layer: This layer consists of 64 units and applies the ReLU activation function. It introduces nonlinearity into the model and helps in learning complex patterns in the data.
- Dense Layer (Output Layer): This layer has a single unit with a sigmoid activation function, which outputs a probability between 0 and 1, representing the sentiment prediction for the input sequence.

Overall, this model utilizes two LSTM layers with bidirectional processing to capture information from the sequence in both directions. The dense layers aid in further processing and making predictions based on the learned representations. After training both models, their predictions are combined using a meta-classifier that employs logistic regression with parameters l2, solver='saga', and C=1.

Table III SVM + LSTM model results

SVM - LSTM model				
Train-Test 70%-30%	*Bi-grams*		*Uni-grams*	
	Stemmed	*Not stemmed*	*Stemmed*	*Not stemmed*
Database 1	0,897	**0,898**	0,896	**0,898**
Database 2	0,939	**0,948**	0,936	**0,942**
Train-Test 80%-20%	*Bi-grams*		*Uni-grams*	
	Stemmed	*Not stemmed*	*Stemmed*	*Not stemmed*
Database 1	**0,900**	0,897	0,899	**0,902**
Database 2	0,940	**0,949**	0,935	**0,943**

From Table III, it can be inferred that, although with a small difference in results, the 80%-20% data split for training and testing yields higher final results. Additionally, it's noteworthy that, unlike other methods,

the combination of SVM and LSTM, where the data is not stemmed, produces higher results compared to cases where word stems are removed. This trend holds for both databases, whether working with single words (uni-grams) or combinations of two words (bi-grams).

In the proposed third model, the same SVM and LSTM are combined, with the LSTM incorporating a personalized attention mechanism, representing a form of content-based attention. Specifically, the approach implements an attention mechanism with the following characteristics:

- The attention mechanism is implemented as a personalized layer in Keras called `Attention`.
- The mechanism employs a neural network with a direct pass to compute attention scores between query vectors (hidden states) and key vectors (encoded input sequences).
- The attention scores are transformed using the 'softmax' function to obtain attention weights.
- The context vector is computed as the weighted sum of value vectors (encoded input sequences) with the attention weights.
- The context vector is then used in the model's predictions.

This type of attention mechanism is similar to an additive attention mechanism, where attention scores are computed by passing query and key vectors through a neural network layer. The mechanism allows the model to focus on different parts of the input sequence depending on their relevance to the current context.

Table IV – SVM+LSTM model +attention mechanism results for Database 1

SVM - LSTM model 70%-30% Database 1				
Activation Function	*Bi-grams*		*Uni-grams*	
	Stemmed	*Not stemmed*	*Stemmed*	*Not stemmed*
activation='relu' activation='sigmoid'	**0,9063**	0,9028	0,9005	0,8978
activation='softmax' activation='sigmoid'	**0,9053**	0,9013	0,8998	0,8978
activation='softmax' activation='softmax'	**0,9052**	0,9039	0,8998	0,9000
SVM - LSTM model 80%-20% Database 1				
Activation Function	*Bi-grams*		*Uni-grams*	
	Stemmed	*Not stemmed*	*Stemmed*	*Not stemmed*
activation='relu' activation='sigmoid'	**0,9075**	0,9051	0,9062	0,9017
activation='softmax' activation='sigmoid'	**0,9073**	0,9085	0,9026	0,9041
activation='softmax' activation='softmax'	**0,9055**	0,9069	0,9017	0,9014

Table V - SVM+LSTM model +attention mechanism results for Database 2

SVM - LSTM model 70%-30% Database 2				
Activation Function	*Bi-grams*		*Uni-grams*	
	Stemmed	*Not stemmed*	*Stemmed*	*Not stemmed*
activation='relu' activation='sigmoid'	**0,9416**	0,9385	0,9378	0,9369
activation='softmax' activation='sigmoid'	**0,9424**	0,9416	0,9375	0,9373
activation='softmax' activation='softmax'	**0,9432**	0,9431	0,9379	0,9369
SVM - LSTM model 80%-20% Database 2				
Activation Function	*Bi-grams*		*Uni-grams*	
	Stemmed	*Not stemmed*	*Stemmed*	*Not stemmed*
activation='relu' activation='sigmoid'	**0,9423**	0,9377	0,9412	0,9371
activation='softmax' activation='sigmoid'	**0,9421**	0,9411	0,9414	0,9351
activation='softmax' activation='softmax'	**0,9433**	0,9426	0,9375	0,9380

From the results presented in Tables IV and V, it can be observed that there is no significant difference in accuracy between the two types of data splits. Once again, the combination of bi-grams and stemming provides the highest results. There is a difference in the combinations of activation functions that yield higher accuracy for the two databases. For Base 1, the combination of **relu** and **sigmoid** generates the highest results at 90.75%. In contrast, for Base 2, this combination is with **softmax** for both layers and achieves an accuracy of 94.33%.

In the fourth developed algorithm, called by the authors **att_SVM+biLSTM+lex_RF**, SVM and LSTM are once again combined, but this time the attention mechanism is integrated into the SVM model. Since attention mechanisms are typically part of neural networks, a specialized class has been defined to combine the SVM method with an attention mechanism. The `fit` method of this class trains a linear SVM model, calculates attention weights based on the SVM coefficients, and stores the trained model. Subsequently, the `predict` method makes predictions using the trained SVM model.

After an in-depth analysis of the results from previous computations, it was found that Logistic Regression, used as a meta-classifier, yielded lower final results compared to using the two models separately. For example, for Binary SVM, it resulted in an accuracy of 0.9399, LSTM had an accuracy of 0.9419, but when they were combined using logistic regression, the final accuracy dropped to 0.9389. This led to the exploration of alternative options for meta-classifiers. Gradient Boosting Classifier and Random Forest Classifier were investigated. It turned out that both of them performed better than Logistic Regression, but the Random Forest Classifier provided the best results.

Additionally, before applying the meta-classifier, a partial lexicon-based approach was integrated, calculating lexicon scores for all comments and combining them with the predictions of the SVM and LSTM models. The combined data were used to train a Random Forest meta-classifier to make final predictions. This approach utilizes both model predictions and lexicon scores to improve classification accuracy. The integrated lexicon-based approach was partially applied only for negative words since the research process uncovered a lexicon containing words with negative connotations only. [2] This use of the lexicon somewhat compensates for the significantly smaller number of negative comments in both databases compared to positive ones.

Table VI - att_SVM+biLSTM+lex_RF for Database 1

SVM - attention- LSTM model 70%-30% Database 1				
Models	*Bi-grams*		*Uni-grams*	
	Stemmed	*Not stemmed*	*Stemmed*	*Not stemmed*
SVM	0.9006	0.8990	0.8981	0.8976
LSTM	0.9008	0.8963	0.9001	0.8977
Random Forest	**0.9125**	0.9051	0.9081	0.9079
SVM - attention- LSTM model 80%-20% Database 1				
Models	*Bi-grams*		*Uni-grams*	
	Stemmed	*Not stemmed*	*Stemmed*	*Not stemmed*
SVM	0.9009	0.8988	0.8993	0.8985
LSTM	0.9008	0.8962	0.9001	0.8974
Random Forest	**0.9105**	0.9055	0.9044	0.9051

Table VII - att_SVM+biLSTM+lex_RF for Database 2

SVM - attention- LSTM model 70%-30% Database 2				
Models	*Bi-grams*		*Uni-grams*	
	Stemmed	*Not stemmed*	*Stemmed*	*Not stemmed*
SVM	0.9403	0.9399	0.9365	0.9339
LSTM	0.9421	0.9396	0.9359	0.9411
Random Forest	**0.9450**	0.9433	0.9414	0.9415
SVM - attention- LSTM model 80%-20% Database 2				
Models	*Bi-grams*		*Uni-grams*	
	Stemmed	*Not stemmed*	*Stemmed*	*Not stemmed*
SVM	0.9397	0.9409	0.9362	0.9346
LSTM	0.9396	0.9368	0.9381	0.9357
Random Forest	**0.9475**	0.9435	0.9452	0.9407

Tables VI and VII highlight an improvement in the final accuracy of the proposed model. For the first time, a result above 91% accuracy was observed for Database 1. Again, applying stemming and using bi-grams gives higher accuracy. But based on the results, no firm conclusion can be drawn as to which data distribution is more appropriate. Thus, for Database 1, the highest accuracy of **91.25%** was achieved with a distribution of 70%-30%, stemming and bi-grams, for Database 2 with a distribution of 80%-20%, stemming and bi-grams, the highest results was **94 .75%.**

The best results using the ensemble method give the following algorithmic steps, which the authors propose for use in the sentiment analysis of comments in Bulgarian language **att_SVM+biLSTM+lex_RF**:

Figure 2 – Algorithmic steps of att_SVM+biLSTM+lex_RF model

IV. CONCLUSION

Experimental studies with different methods and algorithms in machine learning have been carried out with datasets in Bulgarian with different packages of words (comments from different spheres - from customers of hotels and from clients of shops, restaurants, cultural events, and beauty and health centers).

Based on this studies is proposed a new, more accurate algorithm for opinion extraction from comments in Bulgarian. This algorithm is an ensemble method that combines SVM and LSTM, with the attention mechanism included in the SVM model, and an applied Random Forest meta-classifier, combined with a partial lexicon method, which is provoked by and compared with the previously experimental studies of the authors.

REFERENCES

[1]. Brownlee, J.: A Gentle Introduction to Ensemble Learning Algorithms, April 19, 2021 in Ensemble Learning https://machinelearningmastery.com/tour-of-ensemble-learning-algorithms/ (2021)
[2]. https://github.com/AzBuki-ML/public-data/tree/master/polarity_lexicons/hurtlex
[3]. Bahdanau, D.; Cho, K.; Bengio, Y. Neural machine translation by jointly learning to align and translate. arXiv preprint 2014, arXiv:1409.0473
[4]. Munkhdalai, T.; Yu, H. Neural tree indexers for text understanding. In Proceedings of the Association for Computational Linguistics, Vancouver, Canada, 30 July–4 August 2017; Volume 1, p. 11.
[5]. Yang, Z.; Yang, D.; Dyer, C.; He, X.; Smola, A.; Hovy, E. Hierarchical attention networks for document classification. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego, CA, USA, 12–17 June 2016; pp. 1480–1489
[6]. Yin, Y.; Song, Y.; Zhang, M. Document-level multi-aspect sentiment classification as machine comprehension. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, 7–11 September 2017; pp. 2044–2054.
[7]. Lee, G.; Jeong, J.; Seo, S.; Kim, C.; Kang, P. Sentiment classification with word localization based on weakly supervised learning with a convolutional neural network. Knowl. Based Syst. 2018, 152, 70–82.
[8]. Lin, Z.; Feng, M.; Santos CN, D.; Yu, M.; Xiang, B.; Zhou, B.; Bengio, Y. A structured self-attentive sentence embedding. arXiv preprint 2017, arXiv:1703.03130.
[9]. Chen, H.; Sun, M.; Tu, C.; Lin, Y.; Liu, Z. Neural sentiment classification with user and product attention. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, TX, USA, 21 September 2016; pp. 1650–1659.
[10]. Wang, Y.; Huang, M.; Zhao, L. Attention-based lstm for aspect-level sentiment classification. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, TX, USA, 1–5 November 2016; pp. 606–615.
[11]. Liu, G.; Guo, J. Bidirectional LSTM with attention mechanism and convolutional layer for text classification. Neurocomputing 2019, 337, 325–338.
[12]. Petrova, D. (2021) Comparative assay on sentiment analysis on two databases in Bulgarian language, Interdisciplinary Conference on Mechanics, Computers and Electrics, Ankara, Turkey, 27-28 November 2021, ISBN: 978-625-409-707-2
[13]. Petrova, D., Bozhikova V. (2022) Random forest and recurrent neural network for sentiment analysis on texts in Bulgarian language, International Conference on Biomedical Innovations and Applications, Varna, Bulgaria, 2-4 June 2022