International Conference "Risk in Contemporary Economy" ISSN-L 2067-0532 ISSN online 2344-5386
XXIIth Edition, 2021, Galati, Romania,
"Dunarea de Jos" University of Galati, Romania – Faculty of Economics and Business Administration

DOI https://doi.org/10.35219/rce20670532134

Addressing Sentiment Analysis Challenges within AI Media Platform: The Enabling Role of an AI Powered Chatbot

Constantin Avram
Rusu Robert
"Dunarea de Jos" University of Galati, Romania

This paper seeks to classify text with a supervised machine learning algorithm, embedded into an AI powered chatbot. A data set containing tagged texts is used to classify text from IMDb movie review data set, Reviews for Sentiment Analysis - Amazon and Earphones Reviews. The goal is to automatically classify texts into one or more predefined categories. Using supervised learning methods, we developed a model that will use the labelled data set as input. These texts are classified according to syntactic or linguistic characteristics. Research findings outlined that the choice of characteristics for the classification of the sentiments is relevant for leveraging the best possible accuracy, considering Lexicon sentiment, Rules for opinions, Emoticons, Frequency and presence of terms.

Keywords: sentiment analysis, Artificial Intelligence, chatbot, Machine learning, text mining

1. Introduction

Sentiment Analysis (SA) also known as opinion mining involves several areas, such as Natural Language Processing (NLP), web mining and machine learning. This refers to text processing, such as posts or reviews on social networks to identify the emotion behind them or to identify whether they are positive, negative or neutral. For example, when a person wants to buy a product, they have the opportunity to search the internet for reviews and opinions written by other people who have purchased the same product.

Text information can be classified into two main types: facts and opinions. Facts are objective expressions of something. Opinions are usually subjective expressions that describe people's feelings, appreciations and feelings towards a subject.

The analysis of feelings can have various uses, some of the most important being: discovering brands or products present online; checking reviews for a product; customer support and so on.

Sentiment analysis is the process of analyzing a text and classifying opinions. The purpose of this type of analysis is to classify the polarity of a text at the document or sentence level. In addition to identifying polarity, advanced sentiment analysis systems can extract other attributes such as the subject (subject, entity, person or event to which the opinion refers) and the holder. opinion (the person expressing the opinion).

2. Sentiment Analysis and Machine Learning – theoretical considerations

Depending on the learning mode, Machine Learning (ML) techniques can be divided into two categories: supervised learning and unsupervised learning. The first category can be used to automatically group similar items into a collection, in which case no training data sets are required. The second category is used to analyze feeling. Naive Bayes [4], Maximum Entropy [5], [6] and Support Vector Machines (SVM) [7] are the most commonly used techniques in this case.

The Naive Bayes Classifier [4] is a collection of probabilistic algorithms based on the Bayes Theorem and frequently used in ML. The classifier is mainly used in data pre-processing applications due to its ease of calculation. The technique is used to predict the class of a document based on a probability. Attributes play an important role in classification. The classification is made according to each characteristic independently. A disadvantage is that in some scenarios the selected features may not be independent of each other. However, Naive Bayes has the advantage that it is easy to understand and easy to build on a small data set when training, requires less training and applies to both binary problems and multiple classes. The algorithm has applicability for sentiment analysis.

Unlike Naive Bayes, Maximum Entropy [5] makes the classification based on characteristics that are dependent on each other. This is a probabilistic classifier, just like Naive Bayes.

Support Vector Machines (SVM) [7] is a linear algorithm, used mainly in classification problems and which can be applied to several features. Each feature is represented graphically. The value of each characteristic is the value of a certain coordinate. The classification is made by finding the hyperplane that separates the classes very well. The SVM classifier aims to maximize the distance of each data point in this hyperplane using "support vectors" that characterize each distance as a vector.

2.1. Text classification

Text classification is an example of a supervised machine learning task. A data set containing tagged texts is used to classify the classifier. The goal is to automatically classify texts into one or more predefined categories.

Using supervised learning methods, a model is created that will use the labeled data set as input. These texts are classified according to syntactic or linguistic characteristics. The classifier is trained using an ML algorithm. After training, the classifier can be used to predict a label.

The choice of characteristics for the classification of the feeling is important for obtaining the best possible accuracy. Examples of possible characteristics for classifying sentiment would be: Lexicon sentiment, Rules for opinions, Emoticons, Frequency and presence of terms, etc.

Model testing is based on classification. To classify the text, transfer learning is used, where, in the first phase, the training is done on a large corpus. It is then completed on a target corpus. Finally, the classifier is instructed using labeled examples. The following example illustrates the training process.

Figure 1 - Text classification. Training process
Source: [8]

Yu and Dredze [9] propose several methods that combine the architecture of CBOW (Mikolov et al. [10]) and a second objective function that tries to maximize the relationships found within a semantic lexicon. They use both the paraphrase database (Ganitkevitch et al. [11]) and WordNet (Fellbaum [12]) and report that their methods lead to improved language

modeling and semantic similarity tasks. In CBOW, the order of the words in the context does not influence the prediction.

Kiela et al. [13] aim to improve incorporation by increasing the context of a given word while training a skip-gram model (Mikolov et al. [10]).

The use of language representation and machine learning techniques in combination with a supervised classifier is the most widely used approach in the analysis of feelings.

2.2. Deep Learning

Deep Learning [14] is a category of machine learning algorithms inspired by the structure and function of the brain that learns through an artificial neural network.

Recurrent neural networks (NRNs), such as the LONG SHORT-TERM MEMORY (LSTM) network (Hochreiter and Schmidhuber [15]) or GATED RECURRENT UNITS (GRU) (Chung et al. [16]), are a variant of feed networks. -forward which include a memory state capable of learning long distance dependencies. RNNs are useful for text classification tasks (Tai et al. [17]; Tang et al. [18]).

Socher et al. [19] and Tai et al. [17] use Glove vectors (Pennington et al. [20]) in combination with recurrent neural networks and use Stanbank Sentiment Treebank for training (Socher et al. [19]). Because this data set is annotated for feelings at each node of an analysis tree, training and testing is done on these annotated phrases.

Socher et al. [19] and Tai et al. [17] also propose various RNNs capable of making better use of marked nodes that perform better than standard RNNs. However, these models require annotated analysis trees, which are not available for other data sets.

Convolutional neural networks (CNN) have been shown to be effective in classifying text (Santos and Gatti [21]; Kim [22]; Flekova and Gurevych [23]).

Kim [22] uses skip-gram vectors (Mikolov et al. [10]) as input for a variety of convolutional neural networks and tests on seven data sets, including the Stanford Sentiment Treebank (Socher et al. [19]). The most powerful configuration is a single-layer CNN that updates original skip-gram vectors during training.

3. Method

A large amount of data was collected using a semi-automated approach using an AI powered chatbot trained with deep learning algorithms, resulting in a data set of 4,064,337 texts, called fw_senti_text_dataset. Data were extracted from the IMDb movie review data set [29] (50,000 reviews), Reviews for Sentiment Analysis - Amazon [35] (4,000,000 reviews) and Earphones Reviews [36] (14,337 reviews). All those reviews and feelings were stored in CSV files containing the fields: 'feeling', 'text', 'negative', 'neutral', 'positive', 'set'.

Table 1 – Dataset retrieved with the AI powered chatbot

Dataset	No. Msgs	# Train	# Test	# Positive	# Negative	# Neutral
fw_sentiment_set_1 Movie Reviews	50.000	45.000	5.000	25.000	25.000	-
fw_sentiment_set_2 Reviews for Sentiment Analysis **subset de 100K din 4.000K**	100.000	90.000	10.000	51.267	48.733	-
fw_sentiment_set_3 Earphones Reviews Kaggle	14.337	12.903	1.434	9.402	3.432	1503
fw_sentiment	**164.337**	**147.903**	**16.434**	**85.669**	**77.165**	**1503**

Source: "IMDB Dataset of 50K Movie Reviews." [Online]. Available: https://kaggle.com/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews. [Accessed: 29-May-2020]

4. Findings

90% was used for training and 10% for testing in each experiment. The results of the classification using the trained AI powered chatbot are presented in Table 2: Accuracy scores obtained on training on the three subsets of data. As not all sets also contain the neutral label, the experiments were performed only for positive and negative labels.

Table 2: Accuracy scores obtained on training on the three subsets of data

DataSet / Model		fastText	BERT
Movie Reviews	epoch	7	fit_onecycle
	time	1h 4min	5h 4min
	accuracy	**0.881**	**0.9466**
	train samples	45000	
	validate samples	5000	
Reviews for Sentiment Analysis subset (100.000)	epoch	7	fit_onecycle
	time	1h 12min	7h 47min
	accuracy	**0.847**	**0.9538**
	train samples	90000	
	validate samples	10000	
Earphones Reviews Kaggle	epoch	7	fit_onecycle
	accuracy	16min 30s	1h 7min
		0.737	**0.8996**
	train samples	12903	
	validate samples	1434	

Source: authors contribution

The validation of the data set is highlighted in Table 3 and 4, based on representative accuracy scores.

Table 3 : Accuracy scores obtained on validation on the three subsets of data

fastText					BERT				
fastText - set 1	**precision**	**recall**	**f1-score**	**support**	**BERT - set 1**	**precision**	**recall**	**f1-score**	**support**
pos	0.89	0.88	**0.88**	2516	pos	0.96	0.94	0.95	2568
neg	0.88	0.89	**0.88**	2484	neg	0.94	0.95	0.95	2432
accuracy			**0.88**	5000	accuracy			0.95	5000
macro avg	0.88	0.88	**0.88**	5000	macro avg	0.95	0.95	0.95	5000
weighted avg	0.88	0.88	**0.88**	5000	weighted avg	0.95	0.95	0.95	5000
fastText - set 2	**precision**	**recall**	**f1-score**	**support**	**BERT - set 2**	**precision**	**recall**	**f1-score**	**support**
pos	0.86	0.83	**0.84**	4974	pos	0.95	0.95	0.95	4832
neg	0.84	0.86	**0.85**	5026	neg	0.96	0.95	0.96	5168
accuracy			**0.85**	10000	accuracy			0.95	10000
macro avg	0.85	0.85	**0.85**	10000	macro avg	0.95	0.95	0.95	10000
weighted avg	0.85	0.85	**0.85**	10000	weighted avg	0.95	0.95	0.95	10000
fastText - set 3	**precision**	**recall**	**f1-score**	**support**	**BERT - set 3**	**precision**	**recall**	**f1-score**	**support**
pos	0.65	0.48	**0.55**	485	pos	0.89	0.79	0.84	473
neg	0.77	0.87	**0.81**	949	neg	0.90	0.95	0.93	961
accuracy			**0.74**	1434	accuracy			0.90	1434
macro avg	0.71	0.68	**0.68**	1434	macro avg	0.90	0.87	0.88	1434
weighted avg	0.73	0.74	**0.73**	1434	weighted avg	0.90	0.90	0.90	1434

Source: authors contribution

Table 4: Accuracy scores obtained on validation on the final set

fastText					BERT				
fastText - fw	precision	recall	f1-score	support	BERT - fw	precision	recall	f1-score	support
pos	0.69	0.90	0.78	7757	pos	0.95	0.94	0.95	7946
neg	0.88	0.63	0.73	8677	neg	0.95	0.95	0.95	8488
accuracy			0.76	16434	accuracy			0.95	16434
macro avg	0.78	0.77	0.76	16434	macro avg	0.95	0.95	0.95	16434
weighted avg	0.79	0.76	0.76	16434	weighted avg	0.95	0.95	0.95	16434

Source: authors contribution

The training period of the AI powered chatbot involves feeding the bot with different variations of all the possible movie reviews. Table 5 outlines training time for the model.

Table 5: Training times on each data set for each model

DataSet / Model		fastText	BERT
Movie Reviews	process	CPU RAM used 1.10 GB	GPU used 4.38GB RAM used 5.04GB
	epoch	7	fit_onecycle
	time	1h 4min	5h 4min
Reviews for Sentiment Analysis subset (100K)	proces	CPU RAM used 1.10G	GPU used 4.64GB RAM used 6.19GB
	epoch	7	fit_onecycle
	time	1h 12min	7h 47min
Earphones Reviews Kaggle	proces	CPU RAM used 0.77 GB	GPU used 4.81GB RAM used 5.53GB
	epoch	7	fit_onecycle
	time	16min 30s	1h 7min COLAB: 54min 23s
fw_sentiment	proces	CPU RAM used 16,7GB	GPU used 11.21GB RAM used 18GB
	epoch	6	fit_onecycle
	time	3h 18min	16h 5min

Source: authors contribution

The first set has the data divided equally into different labels 50% negative, 50% positive, also the second set has the data divided equally into different labels 50% negative, 50% positive, and the third set 24% negative, 65 % positive and 10% neutral.

The fastText model gets better results on balanced datasets. On the first data set (Movie Reviews), it achieves the best accuracy score (88.10%), and on the Earphones Reviews data set in which the distribution of feelings is not balanced (9,402 positive, 3,432 negative), it reaches a score of 73.70% accuracy.

The BERT model performs much better than the fastText model on all datasets. In general, the model offers an accuracy of 94.66% on the drive data and 95% on the validation data, in the case of the first data set (Movie Reviews), 95.38% on the drive data and 95% on the validation data for the second set, and for the third set it obtains an accuracy of 89.96% on the training data and 90% on the validation data.

5. Conclusions

For the detection of feelings in the text, two models of deep learning networks embedded in a AI chatbot were identified and tested, which presented in the studied literature the best results, namely: fastText and BERT. Data sets for text detection were identified from which Large Movie Review Dataset, Amazon Reviews for Sentiment Analysis, and Amazon Earphones Reviews Kaggle were selected. These sets are available for research and contain attitudes and feelings of customer users who have bought certain products and services and who express their opinion on various social networks. Based on selected sets, our own set was built by combining them and presented in a unique format, a set later used for training the networks. The results were pre-trained network models with an accuracy of 88.10% on fastText and 95.38% on BERT.

Acknowledgement
This work was supported by a grant of the Ministry of Education and Research from Romania, CCCDI – UEFISCDI, project number PN-III-P1-1.2-PCCDI-2017-0800/ 86PCCDI/2018, project name: FUTUREWEB, within PNCDI III.

References

[1] Pang, B.; Lee, L. "A Sentimental Education: Sentiment Analysis Using Subjectivity, Summarization Based on Minimum Cuts." [Online]. Available: https://arxiv.org/pdf/cs/0409058.pdf. [Accessed: May-2020].
[2] "The Porter Stemming Algorithm." [Online]. Available: https://tartarus.org/martin/PorterStemmer/. [Accessed: May-2020].
[3] A. Mitrani, "Feature Engineering with NLTK for NLP and Python," Medium, 18-Oct-2019. [Online]. Available: https://towardsdatascience.com/feature-engineering-with-nltk-for-nlp-and-python-82f493a937a0. [Accessed: 29-May-2020].
[4] "Learn Naive Bayes Algorithm | Naive Bayes Classifier Examples," Analytics Vidhya, Sep. 11, 2017. https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/ [Accessed: May-2020].
[5] Berger A., "A Brief Maximum Entropy Tutorial." [Accessed: May-2020].
[6] Ratnaparkhi, Adwait. (2017). "Maximum Entropy Models for Natural Language Processing." 10.1007/978-1-4899-7687-1_525. [Accessed: May-2020].
[7] S. Patel, "Chapter 2 : SVM (Support Vector Machine) — Theory", Medium, May 04, 2017. https://medium.com/machine-learning-101/chapter-2-svm-support-vector-machine-theory-f0812effc72 [Accessed: May-2020].
[8] Bataa, Enkhbold and Joshua Wu. "An Investigation of Transfer Learning-Based Sentiment Analysis in Japanese." *ACL* (2019). [Online]. Available: https://arxiv.org/pdf/1905.09642.pdf. [Accessed: May-2020].
[9] Yu, Mo & Dredze, Mark. "Improving Lexical Embeddings with Semantic Knowledge." 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014 - Proceedings of the Conference. 2. 545-550. 10.3115/v1/P14-2089. [Accessed: May-2020]
[10] Mikolov, T., et al., "Efficient estimation of word representations in vector space." arXiv preprint arXiv: 1301.3781, 2013. [Accessed: May-2020].
[11] Ganitkevitch, Juri & VanDurme, Benjamin & Callison-Burch, Chris. (2013). "PPDB: The Paraphrase Database." [Online]. Available: [Accessed: May-2020]

[12] Christiane Fellbaum. 1999. Wordnet. Wiley Online Library. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/9781405198431.wbeal1285 [Accessed: May-2020].
[13] Kiela, Douwe & Hill, Felix & Clark, Stephen. (2015). "Specializing Word Embeddings for Similarity or Relatedness." 2044-2048. 10.18653/v1/D15-1242. [Accessed: May-2020].
[14] Y. LeCun, Y. Bengio, and G. Hinton, May 2015, "Deep learning," Nature, vol. 521, no. 7553, pp. 436–444.
[15] Hochreiter, S. and J. Schmidhuber, "Long short-term memory. Neural computation", 1997.9(8): p. 1735-1780. [Accessed: May-2020].
[16] Chung, Junyoung & Gulcehre, Caglar & Cho, KyungHyun & Bengio, Y., (2014). "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling." [Accessed: May-2020].
[17] Tai, Kai & Socher, Richard & Manning, Christoper. (2015). "Improved Semantic Representations from Tree-Structured Long Short-Term Memory Networks." 1. 10.3115/v1/P15-1150. [Accessed: May-2020].
[18] Tang, Duyu & Qin, Bing & Feng, Xiaocheng & Liu, Ting. (2016). "Effective LSTMs for target-dependent sentiment classification." [Online]. Available: https://arxiv.org/abs/1512.01100. [Accessed: May-2020].
[19] Socher, R., et al. "Recursive deep models for semantic compositionality over a sentiment treebank" in Proceedings of the conference on empirical methods in natural language processing (EMNLP). 2013. Citeseer. [Accessed: May-2020].
[20] Pennington, Jeffrey & Socher, Richard & Manning, Christoper. (2014). "Glove: Global Vectors for Word Representation". EMNLP. 14. 1532-1543. 10.3115/v1/D14-1162. [Accessed: May-2020].
[21] Dos Santos, Cicero & Gatti de Bayser, Maira. (2014). "Deep Convolutional Neural Networks for Sentiment Analysis of Short Texts." [Online]. Available: https://www.researchgate.net/publication/274380447_Deep_Convolutional_Neural_Networks_for_Sentiment_Analysis_of_Short_Texts/citation/download [Accessed: May-2020].
[22] Kim, Yoon. (2014). "Convolutional Neural Networks for Sentence Classification". Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing. 10.3115/v1/D14-1181. [Accessed: May-2020].
[23] Flekova, Lucie & Gurevych, Iryna. (2016). Supersense Embeddings: A Unified Model for Supersense Interpretation, Prediction, and Utilization. 2029-2041. 10.18653/v1/P16-1191. [Accessed: May-2020].
[24] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, Aug. 2016, "Bag of Tricks for Efficient Text Classification," arXiv: 1607.01759 [cs]. [Online]. Available: https://arxiv.org/abs/1607.01759 [Accessed: May-2020].
[25] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, "BERT Pre-training of Deep Bidirectional Transformers for Language Understanding", 2018. [Accessed: May-2020].
[26] Manish Munikar, Sushil Shakya, Aakash Shrestha, "Fine-grained Sentiment Classification using BERT", 2019. [Accessed: May-2020].
[27] M. Schuster and K. Nakajima, "Japanese and Korean voice search," in 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2012, pp. 5149–5152. [Accessed: May-2020].

[28] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. (2011). Learning Word Vectors for Sentiment Analysis. The 49th Annual Meeting of the Association for Computational Linguistics (ACL 2011). Available: https://ai.stanford.edu/~amaas/data/sentiment/. [Accessed: 29-May-2020].

[29] "IMDB Dataset of 50K Movie Reviews." [Online]. Available: https://kaggle.com/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews. [Accessed: 29-May-2020].

[30] "Stanford Sentiment Treebank v2 (SST2)." [Online]. Available: https://kaggle.com/atulanandjha/stanford-sentiment-treebank-v2-sst2. [Accessed: 29-May-2020].

[31] "The OpeNER project." [Online]. Available: https://www.opener-project.eu/project/. [Accessed: 29-May-2020].

[32] O. Uryupina, B. Plank, A. Severyn, A. Rotondi, and A. Moschitti, "SenTube: A Corpus for Sentiment Analysis on YouTube Social Media," 2014.

[33] P. Nakov, S. Rosenthal, Z. Kozareva, V. Stoyanov, A. Ritter, and T. Wilson, "SemEval-2013 Task 2: Sentiment Analysis in Twitter," in Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013), Atlanta, Georgia, USA, 2013, pp. 312–320.

[34] "Text Classification Datasets – Xiang Zhang's Google Drive dir." [Online]. Available: https://drive.google.com/drive/folders/0Bz8a_Dbh9Qhbfll6bVpmNUtUcFdjYmF2SEpmZUZUcVNiMUw1TWN6RDV3a0JHT3kxLVhVR2M. [Accessed: 29-May-2020].

[35] "Amazon Reviews for Sentiment Analysis." [Online]. Available: https://kaggle.com/bittlingmayer/amazonreviews. [Accessed: 29-May-2020].

[36] "Amazon Earphones Reviews." [Online]. Available: https://kaggle.com/shitalkat/amazonearphonesreviews. [Accessed: 29-May-2020].

[37] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving Language Understanding by Generative Pre-Training," p. 12.