Transactions on Engineering and Computing Sciences - Vol. 11, No. 2
Publication Date: April 25, 2023
DOI:10.14738/tecs.112.14328.
Sargsyan, S., & Hovakimyan, A. (2023). Creating a Sentiment Analyzer for Text Messages. Transactions on Engineering and Computing Sciences, 11(2). 53-60.

SERVICES FOR SCIENCE AND EDUCATION

Creating a Sentiment Analyzer for Text Messages

Siranush Sargsyan
Yerevan State University, Yerevan, Armenia

Anna Hovakimyan
Yerevan State University, Yerevan, Armenia

ABSTRACT

The work is devoted to the use of Logistic Regression and Neural Network methods to develop a methodology for creating and implementing an analyzer for determining the sentiment of a text in Armenian. Based on a selection of data from social networks, this analyzer determines the tone of the message entered by the user. Note that such work with Armenian texts is carried out for the first time. Sentiment analysis of the text helps to form an opinion about the content of the message without reading the entire text, freeing the user from unnecessary work and time. Sentiment analyzer based on various machine-learning methods for text classification has been created. The paper presents the results of a comparative analysis of the models underlying this analyzer. The created analyzer is used in processing the results of a social survey of students both to determine the students' opinion about the relevance of teaching a given subject and to assess the quality of teaching a subject by a teacher. It is also supposed to use the analyzer to determine the psychological state of the student before the exam in order to adapt the exam to student.

Keywords: sentiment analysis, machine learning, logistic regression, neural network, classifier, social networks.

INTRODUCTION

Recently, thanks to the spread and ubiquitous use of the Internet, the task of analyzing the sentiment of a text is relevant and has gained wide interest. It is the task of Sentiment Analysis [1]. The purpose of Sentiment Analysis is to identify sentiments expressed in textual content. The Sentiment Analysis technique can be used to determine, for example, people's opinion of the quality of a particular product, to analyze the results of a public opinion poll, to determine the sentiment of a film or article review, to determine the quality of a subject's teaching, etc. Sentiment Analysis allows researchers to solve a wide range of problems, such as forecasting the stock market and election results, determining reactions to certain events or news, and so on. Determining the tone of text messages is able to identify and analyze the emotional assessment of the author in relation to the objects referred to in the text. This makes it possible to learn about the author's attitude to any phenomenon, product, service [1, 2].

Sentiment Analysis can be seen as an alternative to traditional surveys. These methods allow not only to show the current attitude to the object of study, but also to predict changes in attitude to the object. For example, in the USA there is a project called Pulse of the Nation, whose

goal is to determine the mood of the citizens of the country during the day by researching and analyzing their records in the popular social network Twitter [2].

Since the beginning of the 2010s current trends in sentiment analysis are its use for social media and multilingual tone analysis. Until now, such works have been occupied by research outside of English-language texts. The reason for this is most often the lack of available thesauri in national languages for tonal analysis of texts.

Until 2014, there were only 12 publicly available non-English lexicons for tone analysis. In 2014, an attempt was made to create dictionaries for other languages. The degree of convergence of the result with the traditional approach was below 50%, and the proposed dictionaries are practically not used.

In 2016-2018 many works have appeared in which texts in the main European languages are analyzed. But in general, the authors do not develop dictionaries for each of the languages, but use machine translation of negative and positive lexemes from available English thesauri. This, of course, reduces the quality of the analysis [3].

Despite the prospects of this direction, it is not yet so actively used in text information processing systems. The reasons are the difficulty of identifying emotional vocabulary in texts, the imperfection of existing text analyzers, and dependence on the subject area. Therefore, the improvement and development of new methods for sentiment analysis based on machine learning is an urgent task [4].

The purpose of this work was to create a sentiment analyzer for Armenian texts from social networks in accordance with their emotional characteristics. Two text classification methods were applied: Logistic Regression and Neural Networks to create a hybrid analyzer model. Various experiments were carried out to assess the quality of this analyzer.

We, the authors of the presented article, have been engaged in the educational process for many years. During the exams, students are in different psychological situations. Before the exam, with the help of a sentiment analyzer, classifying them into different categories using special questionnaires, you can balance their condition and conduct the exam in an adaptive way. Work is underway to create special extended questionnaires to classify students according to their emotional state.

STATEMENT OF THE PROBLEM

The task of analyzing the sentiment of texts has many effective methods for solving, each of which has its own characteristics. To solve these problems using information technology, Natural Language Processing (NLP) methods are used [1, 4].

For a long time, linear models such as support vector machines, logistic regression, etc. have been used in the mainstream NLP methods. Recently, other machine learning approaches have taken over. In particular, neural network models are successfully applied in the field of natural language processing [4,6].

This paper discusses the tools and methodologies that underlie logistic regression and recurrent neural networks [5,6,8]. It is assumed that the models under consideration are able to learn how to classify texts. Many of them learn when they are presented with examples with correct answers.

The classic method for determining the tone of a text is the dictionary method. It is based on dictionaries of emotionally colored words. Each term from the dictionary is assigned its own weight, either manually or with the help of third-party tools.

There are different methods for determining tonality using dictionaries. For example, each term in the dictionary is evaluated as a positive or negative sentiment. The positive coloring of the text is formed based on the weight of the term included in the text with the maximum positive coloring. Negative coloring is also determined. In the absence of a term in the dictionary, it is assigned a minimum positive weight [10,11].

The main drawback when using dictionary methods is the procedure for compiling dictionaries of terms and specifying their weights, since almost always the value of the weight of a term depends on the chosen subject area. These methods do not allow creating a universal dictionary of terms, since their weight in different subject areas can differ significantly or be opposite. The compilation of language corpora has its own characteristics and difficulties. This is typical for those languages that are relatively rarely used on the Internet (for example, the Armenian language).

For our implementation of the models, the compilation of the corpus of the Armenian language was especially time-consuming. The source of information for the Armenian texts was the social networks Twitter and Facebook. We have developed and implemented a scheme for compiling dictionaries of terms and indicating their weights based on these texts.

At the next stage, features are selected, for which it is determined whether they are present in the text or not. For text classification, words, pairs of words, whole phrases, etc., occurring in it, are considered as features. In the constructed analyzer, words expressing emotions are used as features.

DESCRIPTION OF THE DOCUMENT CLASSIFICATION PROBLEM

The classification of texts is one of the main tasks of the search engine, the purpose of which is to include a text in a particular class based on its content. This classification in this case is performed by the sentiment analyzer [4, 5].

Assume that we have finite sets of classes $C = \{c_1 \dots c_{|C|}\}$, documents $D = \{d_1 \dots d_{|D|}\}$ - and the desired function Φ, which for arbitrary pair <c, d>, where c ∈ C, d ∈ D, determines whether document d is in class C or not. $\Phi: D \times C \rightarrow \{0,1\}$.

The task of classification is to find the function Φ', approximating the function Φ. This function Φ' is the classifier function.

At the initial stage of machine learning, a group of documents is allocated $= \{d_1 \dots d_{|Q|}\} \subseteq D$. The Φ function is known for an arbitrary pair $<d_i, c_i> \in Q \times C$.

The subset Q of documents is divided into two disjoint sets: training and testing sets. With the help of the training set $Tr = \{d_1 \dots d_{|Tr|}\}$ the classifier Φ' is determined, and $Te = \{d_{|Tr|+1} \dots d_{|Q|}\}$ is a set of documents on which the classifier Φ' is tested.

Each text document is given to the classifier as an input, and the resulting result $\Phi'(d_i, c_i)$ is compared with the value of the function $\Phi(d_i, c_i)$. The more matches, the more efficient the classifier.

- There are two types of classifications, depending on the value of functions Φ':
- Precise classification – $\Phi': D \times C \rightarrow \{0,1\}$
- Probabilistic classification - $\Phi': D \times C \rightarrow [0,1]$

Thus, in the case of precise classification, an arbitrary pair (d, c) corresponds to the value 0 or 1, that is, the document belongs to this class or not. In the case of probabilistic classification, an arbitrary pair (d, c) corresponds to a number from the interval [0, 1], which indicates the probability that document d belongs to class C.

The main stages of document classification are indexing of documents, selection and training of a classifier, evaluation of the results of the classifier.

During indexing, documents are converted to the same format. Since the volume of documents is quite large, it is necessary to remove words from them that do not contain emotional coloring [10,11]. These can be supporting verbs, links, and more. At the next stage, the documents are converted into vector format, that is, the document is represented as a vector of weights. Then the vectors are normalized.

In the presented work, the effectiveness of the implemented classifiers is evaluated by the most common F1 metric, which includes two main concepts: precision and recall. These values are easily obtained using the confusion matrix [12].

APPLICATION OF LOGISTIC REGRESSION

The analyzer is based on the method of logistic regression. The choice of the method is due to high accuracy rates in the analysis of short messages in English and Russian, which is relevant for the task at hand. As is known, logistic regression is used to predict the probability of an event occurring based on the values of a set of features [4,5].

In our case, the event is the presented text, and the features are the words in the text. A dependent variable y is introduced, which takes one of two values (0-event did not occur, 1-event occurred), and a set of independent variables (called features $x_1, \dots x_n$), based on the values of which it is required to calculate the probability of accepting value of y. The model assumes that the probability of an event y=1 is

$$P\{y = 1 \mid x\} = f(z),$$

where $z = w^T x = w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_n x_n$, x and w are vectors of values of independent variables and regression parameters (weights), respectively, and $f(z)$ is the logistic function

$$f(z) = \frac{1}{1+e^{-z}}$$

Since y takes only the values 0 and 1, the probability of taking the value 0 is

$$P\{y = 0 \mid x\} = 1 - f(z) = 1 - f(\mathrm{w}^T x)$$

To select the parameters $\mathrm{w}_1, \ldots \mathrm{w}_\mathrm{n}$, a training set is defined, consisting of sets of values of independent variables and the corresponding values of the dependent variable y. So, this is a set of pairs $\left(\mathrm{x}^{(1)}, \mathrm{y}^{(1)}\right), \ldots, \left(\mathrm{x}^{(\mathrm{m})}, \mathrm{y}^{(\mathrm{m})}\right)$, where $\mathrm{x}^{(\mathrm{i})} \in \mathrm{R}^\mathrm{n}$ is the vector of values of independent variables, $\mathrm{y}^{(\mathrm{i})} \in \{1,0\}$ is the value of y corresponding to it.

Usually, the maximum likelihood method is used, according to which the parameters $\mathrm{w}_1, \ldots \mathrm{w}_\mathrm{n}$ are selected to maximize the value of the likelihood function on the training sample. In the model, the weight coefficients w is chosen so that the probability y for the input data x from the training sample would be the largest.

$$\hat{w} = \arg\max_w \log P(y^j \mid x^j)$$

For the entire training set, the weight vector is chosen as follows:

$$\hat{w} = \arg\max_w \sum_j \log P(y^j \mid x^j)$$

The objective function L that we are trying to maximize is the following [8]

$$L(w) = \sum_j \log P(y^j \mid x^j)$$

Finding a vector w that maximizes this function is an optimization problem that has different solution methods, such as the gradient boosting method [7]. Taking into account the weights of features in the text, the maximum probability of the text belonging to a particular class is determined.

To implement the model, the Logistic Regression class of the sklearn library was used. First, the text data and their corresponding tones are loaded. The input data was digitized in two ways: Bag of words and word2vec [10, 11]. Based on the input text data, a dictionary of features was built, and with the help of the CountVectorizer class, a vector representation of texts was built. Then, with the help of the analyzer, which receives a vector representation of the training data and the corresponding sentiment (positive, negative), the model is trained. After training the

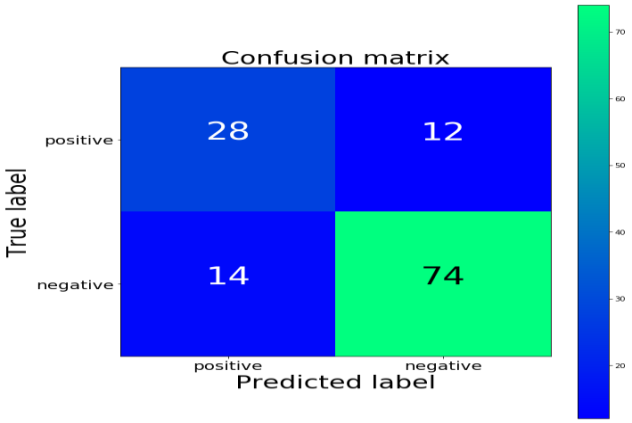
model, it is tested on testing data using their vector representation.

When digitizing texts using the Bag of words method, the following results were obtained on test data: **Accuracy = 0.789, Precision = 0.784, Recall = 0.789, F1 = 0.785.**

When the input data was digitized using the word2vec method, the better results were obtained: **Accuracy = 0.812, Precision = 0.819, Recall = 0.812, F1 = 0.815.**

Accuracy is the proportion of documents for which the classifier made the correct decision. Precision and recall are metrics that are used in evaluating most information extraction algorithms. The F-measure is the harmonic mean between precision and recall [12].

In this case, for the confusion matrix, we got



As can be seen from the figure, out of 40 positive and 88 negative records, 42 records are classified as positive and 86 as negative.

NEURAL NETWORK USAGE

There are many works on the use of neural networks in various tasks of processing texts in natural language, however, the use of various approaches and methods, in particular, depends on the language of the text, which justifies their relevance [6].

For training, a labeled sample was compiled containing a set of some texts and their corresponding emotional assessments. The sample consists of messages from social networks: Twitter -500 texts, Facebook -200 texts. Each selected message was given an emotional rating.

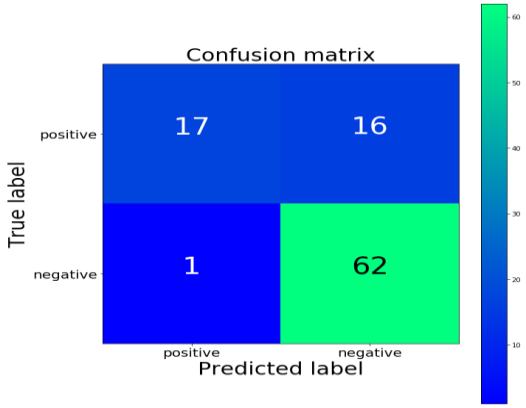
To solve the problem, we used the LSTM (Long Short-Term Memory) network, one of the types of recurrent networks [8]. The network was built using the Tensorflow library methods. This library is quite efficient for complex calculations and can run on the GPU (Graphics Processor Unit), which, unlike the CPU, allows you to perform complex calculations several times faster. The Dropout technique was used to avoid the problem of overfitting [5].

The steps to implement an LSTM network are as follows.

- Text data and their tonalities are loaded at startup. Then symbols and non-essential words were removed (stop words).
- The Bag of Words model was used for vector text representation.
- With the transformation of text data, they were divided into two parts: data for training and testing (80% and 20%).

A recurrent neural network has been created. The neural network parameters were chosen experimentally. After training the network and testing on the test data, the following results were obtained: **Accuracy = 0.812, Precision = 0.810, Recall = 0.812, F1 = 0.811.**

In this case, for the confusion matrix, we got:



As can be seen from the figure, out of 33 positive and 63 negative records, 18 records are classified as positive, and 78 as negative.

COMPARATIVE CHARACTERISTICS OF MODELS

The table below shows the results of implementing a logical regression and a neural network model. For the data used, as can be seen, a better result was obtained for the logistic regression model than for the neural network.

Classifier	Accuracy	Precision	Recall	F1 measure	Training time (sec.)
Logistic regression (Bag of Words)	0.773	0.772	0.773	0.773	0.005948
Logistic regression (word2vec)	0.812	0.819	0.812	0.815	0.042347
Recurrent neural network	0.812	0.810	0.812	0.811	2.53262

CONCLUSION

The task of determining sentiment is complex and the results obtained, of course, do not completely solve the problem of analyzing texts for different subject areas. However, such methods of analysis can become a step towards the creation of automated systems for researching sentiment in a particular subject area. To solve the creation of a sentiment analyzer, Armenian texts were collected and developed from the social networks Twitter and Facebook (twitter-corpus.scv and facebook-corpus.scv). On their basis, software tools have been developed for loading and analyzing the tone of texts/questionnaires entered by the user. The use of the created sentiment analyzer in the educational process improves the learning process by updating the curriculum with subjects demanded by students. In the future, based on the study of methods of content analysis of the emotional coloring of Armenian texts, it is supposed to find a general pattern for processing Armenian texts from different subject areas.

References

[1] Tedmori S., Awajan A., Sentiment Analysis Main Tasks and Applications: A Survey. Journal of Information Processing Systems, 2019, Vol. 15, no. 3, P. 500–519. DOI: 10.3745/JIPS.04.0120.

[2] Pang B., Lee L. Opinion Mining and Sentiment Analysis. Foundations and Trends in Information Retrieval, Vol. 2, No. 1–2, 2008, pp. 135.

[3] Y. Chen, S. Skiena, Building Sentiment Lexicons for All Major Languages. Proceedings of the 52 nd Annual Meeting of the Association for Computational Linguistics, Vol. 2, 2014, pp. 389.

[4] Bo Pang, Thumbs up? Sentiment Classification using Machine Learning Techniques. Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), Philadelphia, July 2002, pp. 417-424.

[5] Underfitting and Overfitting in Machine Learning https://www.geeksforgeeks.org/underfitting-and-overfitting-in-machine-learning/

[6] Denny Britz Introduction to RNNs http://www.wildml.com/2015/09/recurrent-neural-networks-tutorial-part-1-introduction-to-rnns/

[7] Gradient Descent https://en.wikipedia.org/wiki/Gradient_descent

[8] S. Hochreiter, J. Schmidhuber, Long Short-Term Memory. Neural Computation 9(8), pp.1735 - 1780, 1997.

[9] Lai S., Xu L., Liu K., Zhao J. Recurrent Convolutional Neural Networks for Text Classification, AAAI, pp. 2267 – 2273, 2015.

[10] Zhang Y., Jin R., Zhou Z.H. Understanding bag-of-words model: a statistical framework, International Journal of Machine Learning and Cybernetics, Vol. 1, 2010.

[11] Word2vec for Armenian https://github.com/YerevaNN/word2vec-armenian-wiki

[12] Leon Derczynski. Complementarity, F-score, and NLP Evaluation Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16), pp. 261–266, 2016.