LSTM Network Learning for Sentiment Analysis

Badiâa Dellal-Hedjazi and Zaia Alimazighi
Faculty of Computer Science, USTHB University, Algiers, Algeria

Keywords: Sentiment Analysis, Natural Language Processing, Deep Learning, RNN, LSTM, CNN.

Abstract: The strong economic issues (e-reputation, buzz detection ...) and political (opinion leaders identification ...) explain the rapid rise of scientists on the topic of sentiment classification. Sentiment analysis focuses on the orientation of an opinion on an entity or its aspects. It determines its polarity which can be positive, neutral, or negative. Sentiment analysis is associated with texts classification problems. Deep Learning (machine learning technique) is based on multi-layer artificial neural networks. This technology has allowed scientists to make significant progress in data recognition and classification. What makes deep learning different from traditional machine learning methods is that during complex analyses, the basic features of the treatment will no longer be identified by human treatment in a previous algorithm, but directly by the deep learning. In this article we propose a Twitter sentiment analysis application using a deep learning algorithm with LSTM units adapted for natural language processing.

1 INTRODUCTION

With the evolution of the web and especially social networks, there is an explosion in quantities of unstructured data. The challenge is the analysis of these data to make decisions or deduce new knowledge. There are several methods of data analysis such as "text-mining" and "data-mining". Text-Mining is a technique for extracting knowledge from documents or texts that are little or not structured using different computer algorithms. Sentiment analysis is the part of text mining that tries to deter-mine the opinions and sentiments present in a text or set of texts. It provides an over-view of public opinion about certain themes. To analyze these large masses of data, it is necessary to collect, store and clean them, then their coding and their analysis. Sentiment analysis is a classification problem. It consists in determining the polarity (positive, negative) of the analyzed texts. Classification problems on large volumes of data require the use of machine learning techniques and particularly deep learning when statistical or linguistic methods become no longer appropriate. In Section 2 of our paper, we present Sentiment analysis and the approaches and techniques of sentiment analysis in particularly deep learning. Section 3 is devoted to some works on sentiment analysis. Section 4 is for presenting our system and Section 5 to implementation and experimentations. We conclude in section 6 with some research prospects.

2 SENTIMENT ANALYSIS

Sentiment analysis is a very active area of research in NLP and AI. Sentiment analysis is an approach that determines the "position" of the individuals studied with regard to a brand or event. It relies on textual resources but can also depend on other elements such as the use of emoticons, voice analysis or facial coding / decoding, etc. (Liu, 2012; Bathelot, 2018; Makrand, 2014; Lambert et al., 2016; Rakotomalala, 2017; Pozzi et al., 2017).

There are two main approaches for sentiment analysis: lexical analysis and machine learning analysis.

2.1 Approaches and Techniques of Sentiment Analysis

Detection and classification face problems that distinguish them from traditional thematic research whose subjects are often identified by keywords. This sentiment can be expressed in a very varied and subtle way and therefore it is difficult to determine whether it is positive or negative. For this, there are

Dellal-Hedjazi, B. and Alimazighi, Z.
LSTM Network Learning for Sentiment Analysis.
DOI: 10.5220/0010964800003179
In *Proceedings of the 24th International Conference on Enterprise Information Systems (ICEIS 2022) - Volume 1*, pages 449-454
ISBN: 978-989-758-569-2; ISSN: 2184-4992

two used approaches, lexical analysis and machine learning approaches.

2.2.1 Lexical Analysis Approach (Linguistic)

The main task in this approach (Linov, Klekovkina, 2012) is the design of lexicons or opinion dictionaries. Their goal is to list as many opinion-bearing words as possible. These words, then, make it possible to classify the texts in two categories (positive or negative) or three (positive, negative and neutral). The quality of classification in this approach depends on the quality of the lexicon.

2.2.2 Machine Learning Approach

This approach consists on representing each comment as a set of variables, and then building a model from text examples whose label is already known. The template is used to assign a class to a new unlabeled comment (Sanders et al., 2018). Machine learning techniques such as SVM (Alessandro, 2016), Bayesian Classifier (Marty, 2016), and others (Herma, Saifia, 2014). They perform better than linguistic methods. These techniques require annotated databases (tedious annotation task). The difficulty of interpreting the learned models and the genericity of the model depends on the data in the learning corpus. The classification of texts in sentiment analysis (Sebastiani, 2012) shows a great precision. However, this precision is obtained only with a representative collection of labeled learning texts and a rigorous features selection. The classifier trained on texts in one field in most cases does not work with other domains (Chabbou, Bakhouche, 2016). Deep learning is making significant progress in data recognition and classification. Traditional machine learning classification algorithms do not perform well in sentiment analysis compared to Deep Learning. The latter is based on neural networks. It has been developed a lot thanks to the evolution of technologies and computing power.

2.3 Deep Learning

Artificial Nural Networks (ANNs) are highly connected networks of elementary processors operating in parallel. Each elementary processor (artificial neuron) calculates a single output based on the information it receives.

In Figure 1, each entry of the artificial neuron x(n) is multiplied by a connecting weight w(n). These products are summed and fed by a transfer function (Wira, 2009).

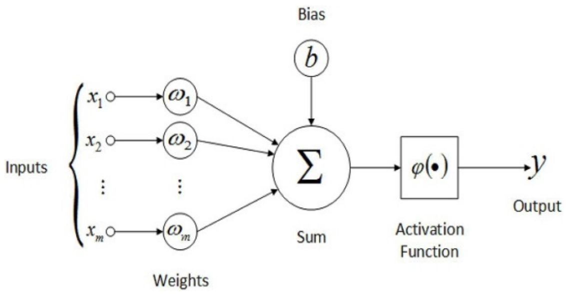

Figure 1: Structure of an artificial neuron (Roserbrock, 2017).

Deep Learning (Deep Neural Networks) belongs to the family of ANN algorithms (Buduma, 2017) (Roserbrock, 2017) (Sugomori et al., 2017) (Skansi, 2018). It is a set of automatic learning methods attempting to model data at a high level of abstraction through articulated architectures of different non-linear transformations. This technique has allowed important and rapid progress in the field of sentiment analysis. Unlike traditional Machine Learning, the essential characteristics of the treatment are no longer identified by human treatment in a previous algorithm, but directly by the Deep Learning algorithm. In these architectures, the input data passes through several computing layers before producing an output. The results of the first layer of neurons serve as input to the calculation of the next layer and so on.

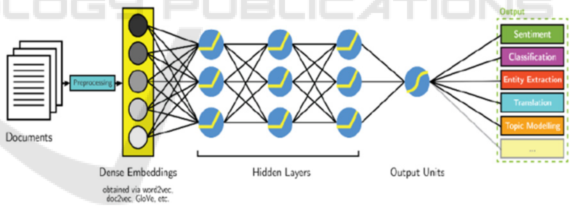

Figure 2: Multi-layer deep neural network (Do et al., 2019).

The first layers of the deep neural network allow to extract simple characteristics that the following layers combine to form increasingly complex and abstract concepts: assemblies of contours in patterns, patterns in parts of objects, parts in objects etc. The more we increase the number of layers, the more the neural networks learns complicated abstract things, corresponding more and more to the way a human reasoning.

There are different types of deep neural networks, multi-layered perceptrons, auto-encoders, CNN (convolutional neural networks), and recursive RNN (recurrent neural networks). RNNs are designed to learn from sequential information where order is important. The RNN performs the same task for each element of a sequence, from which comes the term "recurrent". RNNs are very useful in NLP (Natural Language Processing) tasks (Collobert, Wiston, 2009) because of the sequential dependence of words in any language. For example in the task of predicting the next word in a sentence, the previous word se-quence is of great importance. RNNs calculate memory based on their previous calculations. This memory is used to make predictions for the current step and then forwarded to the next step as input.

2.4 Long Short-Term Memory (LSTM)

When using textual data for prediction, it is important to remember the information long enough and understand the context. RNNs address this problem. These are net-works with loops allowing the information to stay in memory. LSTM networks are a special type of RNN, capable of learning long-term dependencies using LSTM units.

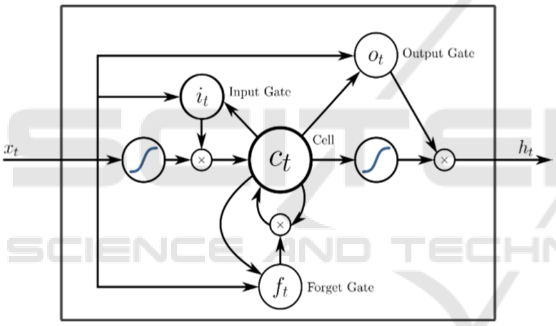

Figure 3: LSTM unit (Roserbrock, 2017).

An LSTM unit is composed of a memory cell, a gate it, an exit gate ot and a gate ft. The input gate controls the extent to which a new value is flowing in the cell, the forget gate controls the extent to which a value remains in the cell and the output gate controls the extent to which the value of the cell is used. LSTM cells can store values (or states) for long periods, unlike standard RNNs.

3 RELATED WORK

The use of Deep Learning for sentiment analysis allows algorithms to understand the structure and semantics of sentences (Marty, 2015). The model is constructed as a representa-tion of the entire sentence based on how words are arranged and interact with each other.

Deep Learning models do not take plain text input: they only work with digital vec-tors. The different units to which one can decompose a text (words, characters or N-grams) are called tokens "tokens", and the process of dividing a text into these tokens is called "tokenization". All text vectorization processes consist of the application of a tokenization scheme, and then associate digital vectors with the generated tokens. These vec-tors fed into a deep neural network.

This vectorization process of the text can be done through "Word2vec". Another model is used for vectorization of text called GloVe "Global Vectors" which is an unsupervised learning algorithm for obtaining vector representations for words. Both Word2vec and Glove are fundamentally similar.

Two models of deep neural networks can be used for sentiment analysis:

- Convolutional Neural Networks (CNN) that apply principles of image processing to the bidimensional sentence vector of a tweet (Marty et al., 2015; Severyin, Moschitti, 2016).
- Recurrent Neural Network (RNN) recursive neural networks that read a number of words specified in the tweet and then output a sentiment probability vector (Wang et al., 2015).

Since recurrent neural networks have memorisation capabilities, they are better suited for the tasks of automatic natural language processing including sentiment analysis where the context of words is important.

In our approach we will explore RNNs based on LSTM units. These have long-term memory capabilities.

4 PROPOSED SYSTEM

Our system consists of the following three main phases and functions:

The Pre-processing Phase: In this phase we prepare our training and test data for the "Large Movie Review Dataset" containing a set of 25000 movie reviews expressing the sentiments and opinions of a group of people towards a set of films they saw. Each film review is stored in a text file, classified by polarity (positive / negative). In order to be able to inject the data of our dataset into the neural network, we must proceed with the vectorization of the text by generating for each word its lexical embedding "word embedding". Each word in each film review is converted into a vector and each sentence into a sequence of vectors. This vectorization process is done by the GloVe model.

Training and Testing Phase: Our data is composed of training data that we pass to the RNN for learning, and test data to test and evaluate the model. The general architecture of the model takes as input the word vectors and the lexicon values for each word from the input data and then the inputs are passed through an LSTM layer with a number of hidden units. The data is vectorized using the GloVe pre-trained vector model. Once the data are prepared, they are injected as inputs to the RNN-LSTM, which requires two very important phases to design a new learning model, the training phase to train the model, and the test phase to evaluate the model.

Prediction (Classification) Phase: The generated representation is then used to de-termine the "positive / negative" polarity of the input text using a fully connected layer with an output Softmax function. The output of the last layer encodes the probability of belonging the text to each class. Once our model is trained, we load the tweet we want to determine its polarity. The tweet is first pre-processed by cleaning the special characters, then vectorized by GloVe and then introduced (the vector) into our trained model, to obtain at output the probability of belonging 0of the tweet to each sentiment class.

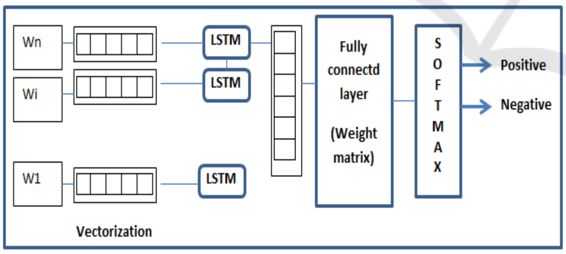

Figure 4: Detailed model architecture.

- The input vector is the word embedding of each word in a given tweet.
- The number of RNN units is chosen during training for optimization purposes. Here we use a monolayer LSTM to avoid overloading the network.
- The weight matrix has as input dimension the size of the RNN, and the number of classes as the output dimension. This means that, taking as input the last LSTM output, we get a vector whose length is equal to the number of classes. This matrix is optimized during the training process.
- The final probability vector is obtained by passing the result of the multiplication of the previous matrix through a softmax function, which converts the component values of this result vector into a representation of probabilities. The predicted tag for the tweet is the component of the output vector with the highest probability.

5 IMPLEMENTATION AND EXPERIMENTS

For the implementation of our application we used essentially Intellij IDEA: It is a Java development environment and Deeplearning4j: library, open-source, distributed for Deep Learning in Java.
We perform the following pre-treatments on the tweet entered by the user:

- Remove websites URLs links, we used the following regular expression: "(http: // (\\ w | \\. | /) + / *) | (Https: // (\\ w | \\ |. /) + / *) "
- Remove all special characters except spaces and punctuation signs, with the fol-lowing regular expression: "[^ a-zA-Z0-9 \\ s!?]""

We get a new tweet that contains only plain text. All the data used are vectorized. For vectorization, we used a vector model pre-trained by GloVe on 1.5 million words. We configured our LSTM recurrent neural network. Next, we create a WordVectors object to load the GloVe pre-trained vector model. We used a DataSetIterator to train and test our data from the Large Movie Review Dataset. Finally, we entered our data and evaluate the modelfor nEpochs times. Each iteration performs the fitting fit method against our trainData training data, and then we create a new evaluation object to evaluate our model using testData test data. The assessment is based on approximately 25,000 movie reviews. Finally, we displayed our evaluation statistics. Once our network is trained, we can make predictions. We load the tweet entered or imported by the user into vector representation, and pass it through the network to predict its probability of belonging to each feeling class. For experimentation, the user can visualize the probability of belonging to the tweet to each "positive / negative" class, as well as the score and training accuracy.

Figure 5: An example of a positive tweet.

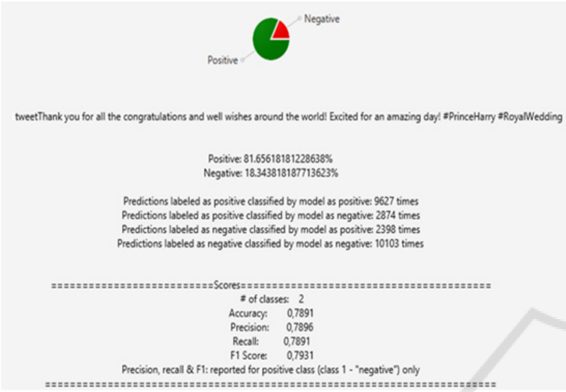

Figure 6: Results for positive tweet.

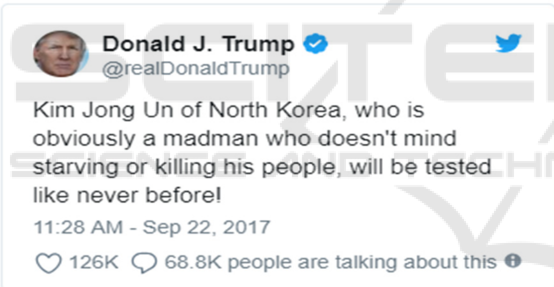

Figure 7: An example of a negative tweet.

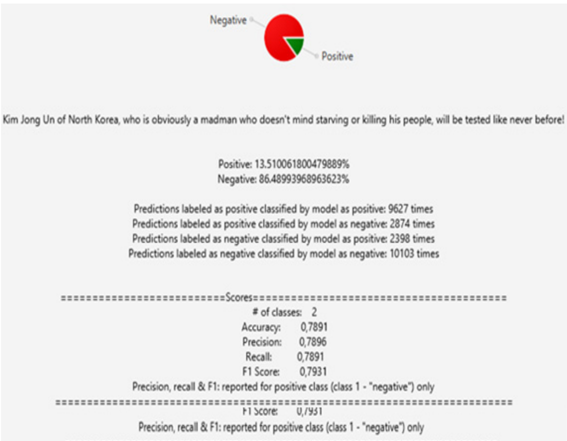

Figure 8: Results for a negative tweet.

When testing our model with a dataset of 3.1 million Amazon reviews, although it is a very large dataset but it gave us an advantage and an effective help for a good learn-ing result, and that because of that richness in words almost 51 thousand of words. The output of our model is. The output of our model is this time modified to have 3 positive, negative and neutral classes. Our results are: Accuracy: 0,91 and loss: 0,22.

6 CONCLUSIONS

This Work is a sentiment analysis application using deep learning. This prototype gave good prediction results. It is a core to exploit and improve for the e-reputation monitoring of companies which will form a platform for decision support as well as a support tool for the recommendation. It can be implemented on a big data platform to better control large volumes of data as well as on fast data (Spark) platforms for real-time and interactive analyses.

REFERENCES

Bathelot, B., (2018), Définitions marketing: Analyse des sentiments.

Herma, S., Saifia, K., (2014). Analyse des Sentiments -cas twitter- Opinion Detection with Machine Lerning. Licence Informatique, université de Ghardaia Algerie.

Makrand, P. A., (2014). Sentiment Analysis: A Seminar Report, SSVPS's B. S. DEORE College of engineering, DHULE.

Linov, P., Klekovkina, D., (2012). Research of lexical approach and machine learning methods for sentiment analysis,- V/Vyatka State Humanities University, Kirov, Russia.

Sebastiani, F., (2012). Machine learning in automated text categorization, ACM, Vol. 34.

Chabbou, F., Bakhouche, S., (2016). Fouille d'opinions méthodes et outils; mémoire master, Université de Tebessa.

Rakotomalala, R. (2017). Fouille d'opinions et analyse des sentiments, Université Lyon 2.

Pozzi, F. A., Fersini, E., Messina, E., Liu, B., (2017). *Sentiment Analysis in social networks*, Morgan Kaufmann Editor.

Liu, B., (2012). *Sentiment Analytics and Opinion Mining*, Morgan & Claypool Publishers.

Sanders, L., Woolley, O., Moize, I., Antulov-Fantulin, N., (2018). Introduction to Sentiment Analytis, Machine Learning and Modelling for Social Networks, D-GESS: Computational Social Science.

Lambert, A., Bellard, G., Lorre, G., Kouki, K., (2016). Analyse de sentiment Twitter, *Proceedings of the 33rd*

International Conference on Machine Learning, New York, NY, USA.
Severyin, A., Moschitti, A., (2016). Twitter Sentiment Analysis with Deep CNN, SIGIR, Chile.
Roserbrock, A., (2017). Deep Learning for computer vision, Pyimagesearch.
Sugomori, Y., Kaluza, B., Suares, F. M., Sousa, A.M. F., (2017). Deep Learning: Pratical Neural Networks with Java, PACKT.
Buduma, N., (2017). Fundamentals of Deep Learning, O'REILLY.
Skansi, S., (2018). Introduction to Deep Learning, Springer.
Wira, P., (2009). Réseaux de neurones artificiels : architectures et applications, UHA Université.
Collobert, R., Wiston, J., (2009). Deep Learning for Natural Language Processing, NIPS Tutorial.
Alessandro E.P., Paolo, V., Antonio, M. , Rivero, J.P, (2016). Artificial Neural Networks and Machine Learning, *25th ICANN*, Barcelona, Spain, September 6–9, 2016 Proceedings, Part II
Marty, J-M., Wenzek, G., Schmitt, E., Coulmance, J., (2015). Analyse d'opinions de tweets par réseaux de neurones convolutionnels. *22ème Traitement Automatique des Langues Naturelles*, Caen.
Wang, X., Liu, Y., Chengjie, S., Wang, B., Wang, X., (2015). Predicting polarities of tweets by composing word embeddings with LSTM. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), vol. 1, pp. 1343–1353.
Do, H. H., Prasad, P., Maag, A., Alsadoon, A., (2019). Deep Learning for Aspect-Based Sentiment Analysis: A Comparative Review, Expert Systems with Applications, Volume 118, 15, Pages 272-299.