Comparative Study of CNN and LSTM for Opinion Mining in Long Text

Submitted: 26th June 2019; accepted: 25th March 2020

Siham Yousfi, Maryem Rhanoui, Mounia Mikram

DOI: 10.14313/JAMRIS/3-2020/34

Abstract: *The digital revolution has encouraged many companies to set up new strategic and operational mechanisms to supervise the flow of information published about them on the Web. Press coverage analysis is a part of sentiment analysis that allows companies to discover the opinion of the media concerning their activities, products and services. It is an important research area, since it involves the opinion of informed public such as journalists, who may influence the opinion of their readers. However, from an implementation perspective, the analysis of the opinion from media coverage encounters many challenges. In fact, unlike social networks, the Media coverage is a set of large textual documents written in natural language. The training base being huge, it is necessary to adopt large-scale processing techniques like Deep Learning to analyze their content. To guide researchers to choose between one of the most commonly used models CNN and LSTM, we compare and apply both models for opinion mining from long text documents using real datasets.*

Keywords: *Deep learning, Long text opinion mining, CNN, LSTM*

1. Introduction

The Web 2.0 has become an official communication space for the press, companies and many governmental or non-governmental organizations. It is also an unofficial communication space, as it allows Internet users to express their ideas, opinions and critics regarding products, services, individuals and special events such as economic or cultural ones. Many organizations are becoming aware of this digital revolution and are implementing new innovative tools to monitor the opinion that the public has built about them and implement, if necessary, preventive or corrective actions.

The Press coverage is an essential element to analyze quantitatively and qualitatively the opinion expressed in the traditional and Web Media. They are realized after a media monitoring and consist of the set of documents related to a brand or a product, following a public-relations operation, a press release, a publicity stunt or an event operation.

The Analysis of the press coverage has many advantages. Indeed, it allows measuring the gain or the lack of reputation of an organization and/or its competitors, regarding a particular action, and identifying good and bad actions in order to take preventive or corrective measures.

Although sentiment analysis has been widely discussed in the literature, most of the published papers focus on social networks. However, compared to the posts and the comments of the social networks, the press articles are a set of large textual documents. Machine learning techniques that have proven their effectiveness for short texts, lead to poor performances for documents, since the knowledge base becomes wider. Moreover, the accuracy of these techniques is going down because it is more likely that a word appears in long text than in a short text and techniques such as BOW (Bag of Words) show low performances [1], [2].

From another point of view, the Deep learning techniques revolutionized the world of data science during the last years. We wondered the contribution the famous learning models to analyze opinions for long text. Thus, through this paper, we propose to apply and compare both CNN (Convolutional Neural Network), and LSTM (long short-time memory) models for opinion mining from press coverage using real world datasets.

The present paper is organized as follows. First, in the section II we provide a general overview about document level sentiment analysis and the deep learning models CNN and LSTM. Then, the section III presents the related works. Sections IV and V present the results and analysis of our benchmark.

2. Background

This section briefly describes the general concepts we are using in this paper, namely Document-Level Sentiment Analysis, Word Embedding techniques and Deep Leaning models CNN and LSTM.

2.1. Document-Level Sentiment Analysis

Sentiment mining or opinion mining of textual data is a field of research that attracted the interest of academia and industry during the last decade, especially with the explosion of data through the mas-

sive use of social media[2]. Several studies target the building of powerful models to analyze sentiments within different fields such as financial forecasting [3], [4] healthcare[5], [6]and others [7], [8].

Although technically challenging, this field is very useful. Indeed, from one hand, it allows companies to discover the opinion of the public regarding their products, and from the other hand, it helps the users to take advantage from the experience of other customers.

There are different levels on which sentiment analysis can be performed according to the level of granularity required [9]:

- Word level sentiment analysis that determines the subjectivity, polarity, and strength of orientation of a word.
- Sentence level sentiment analysis, which determines the subjectivity or the polarity of a sentence.
- Document level sentiment analysis. In fact, at the document level, the objective of sentiment analysis is to assign a global opinion expressed in a document and determine if this opinion is positive or negative. Generally, the whole document is supposed to express the same opinion.

2.2. Word Embedding Techniques

Word embedding is a method that aims to learn the representation of words, by using a vector of real numbers, which facilitates the semantic representation. Two main techniques are used:

- Word2Vec is an unsupervised neural network model that produces word embedding according to the words meaning [10]. Similar words are grouped in a vector space, which preserves the semantic relationship between words.
- Doc2Vec is an extension of Word2Vec developed by Le and Mikolov [11] that deals with the whole document instead of single words. The model creates a numerical representation of the document in order to determine the meaning of a word and to find similarities between documents.

2.3. Bag-of-Words (BoW) Model

The bag-of-words (BoW) model considers documents as a bag of words. It is mainly used to generate textual representations in the NLP (Natural Language Processing) and text mining. However, this model ignores the order of the words. Thus, two documents containing the same words are considered as similar. Several techniques based on neural networks have been proposed to generate dense vectors representing both semantic and syntactic properties of words [12].

2.4. Deep Learning Models

Deep learning is a subset of artificial intelligence that uses algorithms to build models that mimic human behavior. This method is based on artificial neural networks. It had a great success in the field of image recognition, natural language processing and speech recognition.

The artificial neuronal network represents a set of neurons, each receiving an input value with a certain weight. Then, a combination function allows the comparison of the inputs sum of the neuron. And, finally, an activation function captures the difference and compares it to a certain threshold to choose the output and ensure transmission to other artificial neurons[13].

The activation function is an increasing and differentiable function that takes as a parameter the weighted sum of the "x" entries multiplied by the corresponding weights "Wt". The most common functions are the sigmoid function, the hyperbolic tangent function (Tanh), the Softmax function and the rectified linear function (ReLU).

The sigmoid function allows having an output range between zero and one.

It is expressed as follows:

$$\text{Sigmoïde (Wt.x)} = \frac{1}{1+e^{-Wt.x}} \tag{1}$$

The hyperbolic tangent function provides an output range of -1 and 1.

It is expressed as follow:

$$\text{Tanh (Wt.x)} = \frac{e^{Wt.x} - e^{-Wt.x}}{e^{Wt.x} + e^{-Wt.x}} \tag{2}$$

The rectified linear function allows having an output with a threshold of 0 when the input is less than 0.

It is expressed as follows:

$$\text{ReLU (Wt.x)} = \text{Max (0, Wt.x)} \tag{3}$$

An artificial neural network is a set of neurons assembled and connected between the layers that constitute it. It is composed of three main layers; an input layer, an output layer, and an intermediate layer, which can be hidden.

The following Fig. 1 summarizes the architecture of an artificial neural network:

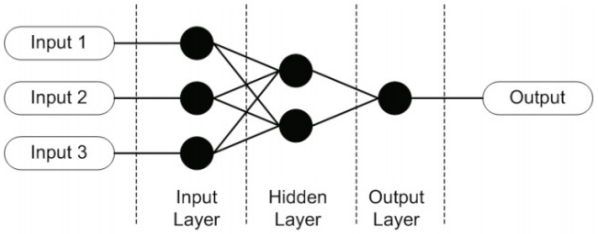

Fig. 1. A simple example of an artificial neural network[14]

2.5. CNN Model

CNN is one of the most successful algorithms used in computer vision. It can detect, segment and recognize objects with excellent noise resistance and variations in position, scale, angle and shape [15]. CNN has also been largely used for NLP tasks such as sentiment analysis, summarization, machine translation, and question answering [16].

The CNN architecture helps to automatically learn the representative characteristics of a given category it receives during the training phase. Subsequently, the CNN seeks these characteristics at the level of a new input data in order to classify it. An example of the architecture of CNN for natural text processing is presented in Fig.2. In fact, it is composed of three different layers namely: input layer, convolution layer and pooling layer.

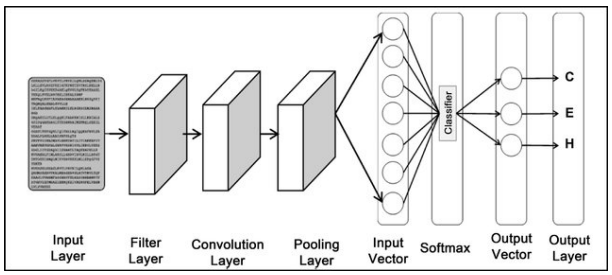

Fig. 2. Example of CNN architecture[17]

Input layer. Each word of the input sentences is represented thanks to the techniques of word embedding in vector $w_i \in R^d$ where d is the dimension of the word embedding. Therefore, the input sentence that contains n words is represented as a matrix with dimensionality $d \times n$.

Convolution layer. Convolution is now the most used concept in deep learning. It defines a mathematical operation that takes two input signals (U_1 and U_2) and returns a new signal (S), where:

$$S(t) = \int_{-\infty}^{+\infty} U1\ (\text{Ţ}).U2(t\text{-Ţ})d\text{Ţ} \qquad (4)$$

It can therefore be seen as an operation where two sources of information are mixed to produce a result. Convolution is performed over the results of the input layer. It combines input data with a filter K k ∈ Rhd which is applied to a window of h words to produce a new feature, called the convolution kernel.

Pooling layer. The pooling layer reduces the inputs by taking only a sample, which helps to minimize the number of parameters and calculations. The most common pooling is the Max-pooling that applies a "max" operation to each region of the filter.

Fully connected layer. This layer attributes to each of the extracted "Features" during convolutions a weight representing the connection strength between the same feature and the corresponding category.

2.6. LSTM (Long Short-Time Memory) Model

It is a network of artificial neurons, where the direction of information diffusion is bidirectional, using an internal memory. The LSTM model is based on back propagation over time and allows prediction with sequential data. It is used whenever there is a sequence of data, and the temporal dynamic that connects the data is more important than the spatial content.

LSTM is a deep learning model intended for long-term dependencies. It uses an internal memory that allows reading, writing and deleting data, according to their importance. The weights learned by the algorithm determine this level of importance.

As represented in Fig. 3, LSTM contains three gates: an Input Gate that allows receiving or not a new entrance, a Forget Gate that ensures the suppression of unnecessary information and an exit door (Output Spoiled).

This architecture helps to solve the problem of the disappearance of the gradient the various gates are analog and allow back propagation.

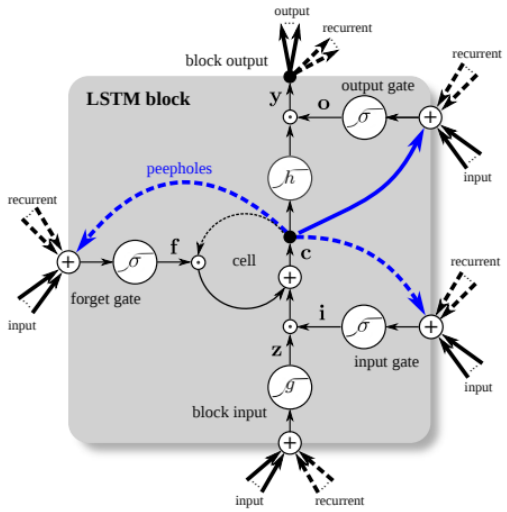

Fig. 3. Detailed description of LSTM architecture [18]

3. Related Works

Different deep-learning-based techniques including CNN, and LSTM are compared in papers [12], [19] and [20].

Yin et al. [1] provide a comparative between different Deep learning models including CNN , GRU (Gated Recurrent Unit) and LSTM among a large number of NLP tasks including sentiment classification. The authors conclude that the performances of the different studied models are very close with a slight overshoot for LSTM. The benchmark targeted social network sentences that contain between 5 and 65 words. As discussed above, these benchmarks used for short text analysis are not necessarily adapted to long texts. In our case, we focus our study on large documents.

Different researches have applied the Deep Learning models for sentiment analysis in documents; Tang et al. [21] propose a new neural network model called User Product Neural Network (UPNN) to capture user-level and product-level information for opinion analysis on documents. Xu et al. [22] present a Cached Short-Term Neural networks (CLSTM) to capture the overall semantic information in long texts. The proposed method divides memory into memory with a low forgetting rate, that captures the global semantic features, and memory with high forgetting rate captures the local semantic features. Yang et al. [23] proposed a hierarchical attention network for document classification. In order to build the representation of the document, the presented model includes two levels of attention mechanisms; word level and sentence level. In fact, it builds a document vector by aggregating important words into sentence vectors and then aggregating important sentences vectors to document vectors that improves performances. These works are applications of Deep Learning models for short text; they did not provide a comparative study to measure the performance and define the optimal model in the context of the sentiment analysis in documents.

4. Experimental Environment

This section aims to present the technical details about the implementation of our benchmark as well as an analysis of results.

4.1. Dataset Description

We have performed our experiments on real datasets collected from web media including electronic newspapers and magazines. The documents were collected from January 2018 to June 2018 in a CSV file with 2275 rows written in French. The CSV file contains the following columns:

- Sector: that represents the context of data, such as agriculture, automobile, healthcare, etc.
- Brand: the brand concerned with the review.
- Media: the name of the journal and magazine that published the data.
- Title: the title of the published content.
- Text: the published content.
- Polarity: it indicates whether the text is positive or negative.

4.2. Technical Environment and Architecture

We have conducted the test on Inter® Xeon® CPU E3-1240 with 3.50GHW, 8,00 GO of memory.

The environment is based on the following tools and libraries:

- MongoDB: this is the NoSQL database where the input data and the results of the analysis are stored.
- Conda: this is the Python-based analysis environment that allows the management of used packages.
- PyMongo: this is the python distribution containing tools that connect to MongoDB.
- Pandas, NumPy, Sklearn: these are Python libraries that provide tools for operations performed on data during processing.
- TensorFlow: this is a library of software dedicated to Deep Learning that provide complex calculations and analyzes. Keras is one of them.
- Keras: the API of neural networks.

4.3. Data Preprocessing

Before applying the CNN and LSTM models, we had to preprocess our initial data in order to make them exploitable. First, we have extracted the two attributes useful for the analysis step namely “text” and “polarity”. Then, using beautiful soup library we performed a cleaning step in order to eliminate spaces, html codes, etc.

Finally, we divided the dataset into three parts, with 70% for the learning base, 22% for the validation base and 8% for the test database.

4.4. CNN Architecture

We adopted the following CNN architecture:

1) Input layer: As explained before, our dataset contains 2275 rows. The maximum number of words contained in a document is 4500. Therefore, the size of the input matrix is 10237500. Since our documents are very long, we choose the Doc2vec model in order to build the embedding vector of the input data since it offers better performances for documents use cases and allows building a general overview about the document.
2) Convolution: The input matrix is very large we choose to perform many convolution steps in parallel in order to reduce the number of parameters. In order to build our convolution layer we were inspired by [15].
   The first convolution step applies 100 bigram filters with kernel = 2.
   The second convolution applies 100 trigram filters with kernel = 3.
   The third convolution applies 100 forgrams filters with kernel = 4.
   The fourth convolution applies 100 fivegrams filters with kernel = 5.
3) Max-pooling: We build each convolution layer followed by a Max-pooling layer. Then the different Max-pooling layers are merged in one output layer.
4) Fully connected Layer
5) The function of activation: Sigmoid
   This function is used to transform the results obtained and to assign the features to the category designating their polarity, whether they are positive or negative.
6) The loss function “binary cross entropy”.
7) The Adam optimization algorithm.

4.5. LSTM Architecture

For the LSTM model, that we designed the following architecture:

- The Word Embedding (WE)
  For this step, we define a set of K vectors representative of each polarity by associating with each word belonging to a polarity a vector Xj in a space of dimension d equal to 10237500.
- The LSTM layers:
  They have the particularity of memorizing the chronological order of words, which is beneficial for long sentences.
- Fully-connected layer:
  This layer provides the connection to the output layer by determining the connection weight between the vectors and their category.
- The Softmax activation function:
  This layer allows converting the vector into probabilities on the polarity that we want to detect. The Softmax function provides an output range from 0 to 1, while the sum of all the outputs is equal to one. We choose this function because our model tries to define the category of each input.
- The loss function "binary cross entropy".
- The Adam optimization algorithm.

4.6. Results and Analysis

Table 1 shows the experimental results for both CNN and LSTM deep learning models for opinion mining from long textual documents. In fact, Fig.4 and Fig.6 show that the training and validation loss are getting closer between CNN and LSTM with the increase of the number of epochs. Also, as shown in Fig. 5 and Fig.7 both models provide good results with a slight outperformance for CNN with 97% of accuracy during the testing step.

Finally, the accuracy of the CNN model improves considerably from the first draft. This is not the case for the LSTM model, since the results improve slowly with the increase of the number of epochs. This is because the CNN implementation uses Doc2Vec model that helped to build the polarity of the whole document. In return, LSTM provides good results, although it's not combined with the doc2vec model. Indeed, unlike the CNN, LSTM model keeps a memory that allows locating a word in context, which can be similar to doc2vec.

Tab. 1. Performance comparison results

Model	Training		Validation		Test	
	Loss	*Accuracy*	*Loss*	*Accuracy*	*Loss*	*Accuracy*
Doc2vec+CNN	5%	97%	16%	95%	10%	97%
LSTM	21%	92%	28%	93%	16%	94%

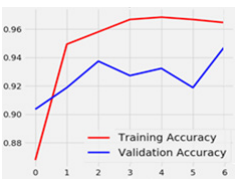

Fig. 4. CNN loss curve according to the number of epochs

Fig. 5. CNN accuracy curve according to the number of epochs

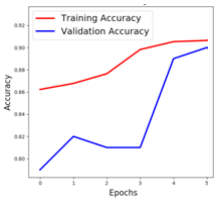

Fig. 6. LSTM loss curve according to the number of epochs

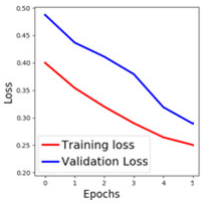

Fig. 7. LSTM accuracy curve according to the number of epochs

5. Conclusion

In this paper, we applied the two famous deep learning models CNN and LSTM for opinion mining from long textual documents. We compared the performances of both models using real-world datasets collected from electronic newspapers. We found that combining Doc2vec and CNN models slightly surpasses LSTM performances.

As a perspective, we are currently looking for other models combinations involving Doc2vec, CNN and LSTM in order to improve performances.

Acknowledgements. The authors gratefully thank Miss Naqachi Hajar, graduate students from the School of Information Sciences in Rabat for her cooperation and invaluable contribution to the validation of this work.

AUTHORS

Siham Yousfi* – SIP Research Team, Rabat IT Center, Mohammed V University in Rabat, Rabat, Morocco, Meridian Team, LYRICA Laboratory, School of Information Sciences, Rabat, Morocco,
email: sihamyousfi@research.emi.ac.ma.

Maryem Rhanoui – IMS Team, ADMIR Laboratory, Rabat IT Center, ENSIAS, Mohammed V University in Rabat, Morocco,
Meridian Team, LYRICA Laboratory, School of Information Sciences, Rabat, Morocco,
email: mrhanoui@gmail.com.

Mounia Mikram – LRIT, Faculty of Science, Mohammed V University in Rabat, Rabat, Morocco,
Meridian Team, LYRICA Laboratory, School of Information Sciences, Rabat, Morocco,
email: mikram.mounia@gmail.com.

*Corresponding author

REFERENCES

[1] W. Yin, K. Kann, M. Yu and H. Schütze, "Comparative Study of CNN and RNN for Natural Language Processing", *arXiv preprint*, 2017, arXiv:1702.01923.

[2] N. C. Dang, M. N. Moreno-García and F. De la Prieta, "Sentiment Analysis Based on Deep Learning: A Comparative Study", *Electronics*, vol. 9, no. 3, 2020, 10.3390/electronics9030483.

[3] S. Sohangir, D. Wang, A. Pomeranets and T M. Khoshgoftaar, "Big Data: Deep Learning for financial sentiment analysis", *Journal of Big Data*, vol. 5, no. 1, 2018, 10.1186/s40537-017-0111-6.

[4] H. Jangid, S. Singhal, R. R. Shah and R. Zimmermann, "Aspect-Based Financial Sentiment Analysis using Deep Learning". In: *Companion of the The Web Conference 2018*, 2018, 1961–1966, 10.1145/3184558.3191827.

[5] R. Satapathy, E. Cambria and A. Hussain, *Sentiment Analysis in the Bio-Medical Domain*, Springer, 2017.

[6] A. Rajput, "Chapter 3 – Natural Language Processing, Sentiment Analysis, and Clinical Analytics". In: M. D. Lytras and A. Sarirete (eds.), *Innovation in Health Informatics*, 2020, 79–97.

[7] M. Kraus and S. Feuerriegel, "Sentiment analysis based on rhetorical structure theory: Learning deep neural networks from discourse trees", *Expert Systems with Applications*, vol. 118, 2019, 65–79, 10.1016/j.eswa.2018.10.002.

[8] L. Li, T.-T. Goh and D. Jin, "How textual quality of online reviews affect classification performance: a case of deep learning sentiment analysis", *Neural Computing and Applications*, vol. 32, no. 9, 2020, 4387–4415, 10.1007/s00521-018-3865-7.

[9] M. M. S. Missen, M. Boughanem and G. Cabanac, "Opinion mining: reviewed from word to document level", *Social Network Analysis and Mining*, vol. 3, no. 1, 2013, 107–125, 10.1007/s13278-012-0057-9.

[10] T. Mikolov, I. Sutskever, K. Chen, G. Corrado and J. Dean, "Distributed Representations of Words and Phrases and their Compositionality". In: *Advances in Neural Information Processing Systems*, vol. 26, 2013, 3136–3144, arXiv:1310.4546.

[11] Q. Le and T. Mikolov, "Distributed Representations of Sentences and Documents". In: *International Conference on Machine Learning*, 2014, 1188–1196.

[12] L. Zhang, S. Wang and B. Liu, "Deep learning for sentiment analysis: A survey", *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 8, no. 4, 2018, 1188–1196, 10.1002/widm.1253.

[13] Y. LeCun, Y. Bengio and G. Hinton, "Deep learning", *Nature*, vol. 521, no. 7553, 2015, 436–444, 10.1038/nature14539.

[14] A. Krenker, J. Bešter and A. Kos, "Introduction to the Artificial Neural Networks", *Artificial Neural Networks – Methodological Advances and Biomedical Applications, IntechOpen*, 2011, 10.5772/15751.

[15] A. Krizhevsky, I. Sutskever and G. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems*, vol. 25, 2012, 1097–1105.

[16] T. Young, D. Hazarika, S. Poria and E. Cambria, "Recent Trends in Deep Learning Based Natural Language Processing", *IEEE Computational Intelligence Magazine*, vol. 13, no. 3, 2018, 55–75, 10.1109/MCI.2018.2840738.

[17] A. Ibrahim and I. Yasseen, "Using Neural Networks to Predict Secondary Structure for Protein Folding", *Journal of Computer and Communications*, vol. 5, 2017, 10.4236/jcc.2017.51001.

[18] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink and J. Schmidhuber, "LSTM: A Search Space Odyssey", *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 10, 2017, 2222–2232, 10.1109/TNNLS.2016.2582924.

[19] Q. T. Ain, M. Ali, A. Riaz, A. Noureen, M. Kamran, B. Hayat and A. Rehman, "Sentiment Analysis Using Deep Learning Techniques: A Review", *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 8, no. 6, 2017, 10.14569/IJACSA.2017.080657.

[20] P. Singhal and P. Bhattacharyya, "Sentiment Analysis and Deep Learning : A Survey", *Center for Indian Language Technology, Indian Institute of Technology, Bombay*, 2016.

[21] D. Tang, B. Qin and T. Liu, "Learning Semantic Representations of Users and Products for Document Level Sentiment Classification". In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, 2015, 1014–1023, 10.3115/v1/P15-1098.

[22] J. Xu, D. Chen, X. Qiu and X. Huang, "Cached Long Short-Term Memory Neural Networks for Document-Level Sentiment Classification", *arXiv preprint*, 2016, arXiv:1610.04989.

[23] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola and E. Hovy, "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*, 2016, 1480–1489, 10.18653/v1/N16-1174.