machine learning & knowledge extraction

MDPI

Article

A CNN-BiLSTM Model for Document-Level Sentiment Analysis

Maryem Rhanoui [1,2], Mounia Mikram [2,3], Siham Yousfi [2,4] and Soukaina Barzali [2]

1 IMS Team, ADMIR Laboratory, Rabat IT Center, ENSIAS, Mohammed V University in Rabat, Rabat 10100, Morocco
2 Meridian Team, LYRICA Laboratory, School of Information Sciences, Rabat 10100, Morocco
3 LRIT Laboratory, Associated Unit to CNRST (URAC 29), Rabat IT Center, Faculty of Sciences, Mohammed V University, Rabat 10100, Morocco
4 SIP Research Team, Rabat IT Center, EMI, Mohammed V University in Rabat, Rabat 10100, Morocco
* Correspondence: mrhanoui@esi.ac.ma

Received: 30 June 2019; Accepted: 23 July 2019; Published: 25 July 2019

Abstract: Document-level sentiment analysis is a challenging task given the large size of the text, which leads to an abundance of words and opinions, at times contradictory, in the same document. This analysis is particularly useful in analyzing press articles and blog posts about a particular product or company, and it requires a high concentration, especially when the topic being discussed is sensitive. Nevertheless, most existing models and techniques are designed to process short text from social networks and collaborative platforms. In this paper, we propose a combination of Convolutional Neural Networks (CNN) and Bidirectional Long Short-Term Memory (BiLSTM) models, with Doc2vec embedding, suitable for opinion analysis in long texts. The CNN-BiLSTM model is compared with CNN, LSTM, BiLSTM and CNN-LSTM models with Word2vec/Doc2vec embeddings. The Doc2vec with CNN-BiLSTM model was applied on French newspapers articles and outperformed the other models with 90.66% accuracy.

Keywords: sentiment analysis; document level; Doc2vec; CNN-BiLSTM

1. Introduction

Opinion or sentiment analysis [1] is a set of linguistic operations belonging to the automatic processing of natural language that apply to digital texts, namely publications and comments from social networks, as well as press articles. Its objective is to identify the sentiment expressed in the text and to predict its polarity (positive or negative) towards a given subject [2].

This analysis is very useful; especially with the emergence of social networks, people start to express their views easily and in the shortest time, which makes the manual processing of this huge number of opinions very difficult.

Machine learning and deep learning models reach high performances when applied to short text, thanks to the abundance of datasets extracted from social networks, and the reduced number of words that facilitate the identification of opinions. However, in the case of a document, the task is more complex because it contains a high number of words and the semantic link between sentences is more problematic.

Researchers have shown a growing interest in the analysis techniques of this large mass of data in order to extract the point of view expressed on a well-defined subject. Deep learning marked the evolution of this analysis given its performance in all that is related to the automatic processing of natural language, namely text summary, word prediction and text classification, it is often used to determine the polarity of different text length levels according to the requested level of granularity.

Sentiment analysis is a popular research topic; several studies have researched and proven the effectiveness of deep neural networks in this task. Indeed, Convolutional Neural Networks (CNN) have achieved great performance in document classification [3]. Long Short-Term Memory Neural Networks (LSTM) is also popular in natural language processing [4]. Individually, convolutional and recurrent models can be considered equivalent in terms of performance [5]. As CNN is a powerful feature-extracting model and is better to be integrated and combined into larger network [6], the challenge now lies in hybridizing and combining these models to take advantage of each other's strengths.

In addition, given the growth of social networks that have become a thriving source of information, research has focused on short text analysis. Document-level sentiment analysis [7] is a more challenging task because the structure, number of words and synthesis of opinions differ considerably from those of the short text.

Thus, the research question that arises is how to propose an effective deep learning model that is adapted to document level processing and classification. To answer this question, our contribution is to propose a CNN-BiLSTM model, which is a combination of convolutional and bidirectional recurrent neural networks for document-level sentiment analysis with Doc2vec Embedding. The model was compared and confronted with the CNN, LSTM, BiLSTM and CNN-LSTM models, and the experiment shows that the CNN-BiLSTM model associated with Doc2vec word embedding outperformed the other models and achieved a 90.66% accuracy in classifying French press articles.

The remainder of this paper is organized as follows. In Section 2, we present the background knowledge. Section 3 presents some related works. Section 4 details the proposed CNN-BiLSTM model. Finally, Section 5 synthesizes and presents the experimental results.

2. Background and Context: Sentiment Analysis

2.1. Sentiment Analysis

Opinion means a judgment, a review or a personal point of view. Each opinion can be positive, negative or neutral. This opinion classification has received a lot of attention lately, since opinion has become an essential element in e-reputation monitoring systems.

2.1.1. Levels of Opinion Analysis

Opinion analysis can be applied on different levels of granularity, namely:

Word level is the analysis that determines the polarity of a word, i.e., if it is a positive, negative or neutral word.

Sentence level is the analysis which determines the polarity of a sentence. It is often used in the analysis of opinion. Since it is specific to social networks, it is a sequence of words which aims at defining an opinion on a subject.

Document level is the analysis that determines the polarity of a document. It is a more difficult level compared to the others, because, when the number of words increases, noise words increase, which distorts learning and complicates the prediction of polarity.

2.1.2. Word Embedding

Word Embedding is a method used in deep learning for the automatic processing of natural language, based on the representation of words in a large corpus by projecting a set of words of size k into a vector space of dimension M such that $M < K$, in order to facilitate the semantic analysis of words and to improve learning performance.

A **Word Vector [8]** is a network of artificial neurons of two layers that have the ability to learn how to represent each word with a real number vector with its semantic features, thus it allows grouping similar words into a single vector.

The Word2vec can learn through two models:

- Continuous Bag of Words (CBOW) is based on the prediction of a word from a context. It can be a single word or a set of words, thus this model has a very strong point because it does not need many resources. This model is based on the calculation of the negative logarithmic probability of a word (w_t) with respect to a context $(\hat{r})$

$$-\log P(w_t|\hat{r})$$

where

$$P(w_t|\hat{r}) = \frac{exp(w_t^T \hat{r})}{\sum_{w \in W} exp(w^T \hat{r})}$$

- Skip-Gram is similar to CBOW except that this model takes a word as input and predicts all the other words as output.

A **Document Vector [9]** is an extension of the Word Vector, which represents the whole of a document in a single digital vector in order to easily identify the similarity between the documents. It can learn the representation of documents through two models, namely Distributed Bag of Words (DBOW), which is equivalent to Skip-Gram in Word2vec, and Distributed Memory (DM), which is equivalent to CBOW.

- Distributed Bag of Words (DBOW) randomly predicts a probability distribution of words in a document from the identifier of a document, in order to create its vector. It does not take into consideration the order of words. During training, the document vector and word weights are randomly initialized and updated using the stochastic gradient descent.
- Paragraph Vector—Distributed Memory (PV-DM), unlike DBOW, predicts a word from the context of the document. It takes a set of words of a paragraph randomly and a document identifier as input and tries to predict a central word.

2.2. Deep Learning

2.2.1. CNN

Convolutional Neural Networks (CNN) are a form of artificial neural networks that can detect information in different positions with excellent accuracy. This model has solved several problems in image processing and automatic natural language processing such as opinion analysis, answers to questions, text summary, etc. It is characterized by a particular architecture to facilitate learning. A convolutional neural network is a multilayer network, so that the output of one layer will be the input of the next layer. It is usually composed of an input, one to several hidden layers and an output.

2.2.2. RNN-LSTM

Recurrent neural network (RNN) is an interconnected and interacting network of neurons, where the neurons are connected by arcs of a weight w. This type of network is very useful in the case of inputs of varying sizes, as well as for time series, namely automatic translation, automatic speech recognition and automatic pattern recognition. The direction of propagation of the information in this type of artificial neural network is bidirectional; it keeps the sequence of data; and it is able to make the connection between an input of long sequences because it is based on a loop thanks to its internal memory.

Long Short-Term Memory Neural Networks (LSTM) were proposed by Hochreiter and Schmidhuber in 1997 [10,11]. They are an extension of RNN able to solve the problem of the vanishing of the gradient thanks to its memory, which makes it possible to read, write and delete the data through three gates: the first allows or blocks the updates (Input Gate); the second disables a neuron if it is not important based on the weights learned by the algorithm, which determines its importance (Forget Gate); and the third is a control gate of the neuron state in the output (Output Gate) (Figure 1).

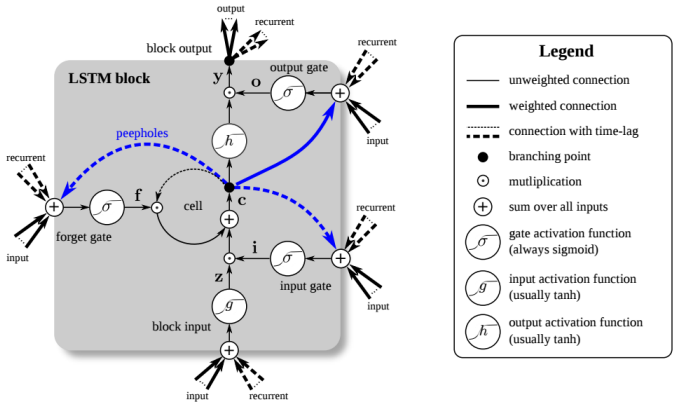

Figure 1. LSTM block [12].

This model has shown its power especially in long-length entries as in the analysis of documents, at it builds the relationship between the different words of the document in order to predict its tone with great precision [13].

2.2.3. RNN-BiLSTM

Bidirectional Long Short-Term Memory (BiLSTM) [14] is a type of recurrent neural networks. It processes data in two directions, since it works with two hidden layers. This is the main point of divergence with LSTM. BiLSTM has proven good results in natural language processing [5,15].

3. Related Works

Sentiment/opinion analysis is a main-spread research subject using machine learning [16] and deep learning [17] models. It has been popularized by tweets analysis, and therefore by short text analysis at sentence-level [18–22]. For deep learning models, this classification requires a fundamental cleaning and pre-processing step (including tokenization, stop-word removal, lowercase conversion and stemming [23]) since data quality has a significant influence on the performance of the deep learning model [24].

Pang et al. [7] first introduced the concept of document-level sentiment analysis. In this section, we present and discuss sentiment analysis approaches at both sentence-level (short text) and document-level (long text).

3.1. Short Text Sentiment Analysis

Dos Santos and Gatti [25] worked on the short text. They proposed a new deep convolutional neural networks that exploits character-to-sentence information to perform sentiment analysis in short texts and validated it with two databases, the Stanford Sentiment Treebank (SSTb), which contains sentences taken from film reviews, and Stanford's feeling on Twitter (STS). They proposed a network named "Character Convictional Convictional Neural Network (CharSCNN)" that uses two convolutional layers to extract the relevant features of words and sentences of all sizes. This model achieved 85.7% precision on SSTb and 86.4% on STS.

In addition, Zhou et al. [26] applied LSTM bidirectional with two-dimensional max pooling on the Stanford Sentiment Treebank (STS) database, such that each vector is represented by a matrix of dimension 2. Thus, they changed the usual use of 2D pooling to sample more relevant features for sequence modeling tasks. In addition, they used 2D convolution to highlight the most important information on the matrix. The BLSTM-2DPooling combination achieved a performance of 88.3% while the BLSTM-2DCNN combination achieved a performance of 89.5% on the SST2 database.

Kim [3] applied a simple CNN with a single convolutional layer using an unsupervised model, on several database, namely movie reviews, STS, subjectivity dataset, TREC query dataset,

and customer reviews. He used a small hyperparameter, which generated very powerful results. He also used different filters sizes and tested several CNN models to extract the important data. He concluded that CNN-static is the model that gave the best performance.

Wang et al. [27] worked on the combination of CNN and RNN for the opinion analysis in the sentences. They wanted to benefit from the advantages of the CNN and the RNN in order to have more precision. They applied the CNN, which is insensitive to the location of the words in the sentence. Then, they used the output of the CNN as input for RNN trained with back-propagation through time.

Yenter and Verma [28] proposed a CNN-LSTM model for opinion analysis from the IMDB database. This work differs from the previous one as they concatenated the results after the application of the LSTM layer. This model achieved 89% accuracy.

Shen et al. [29] proposed a special design that combines the CNN and BiLSTM models for optimal performance. They found that this combination gave an accuracy of 89.7%, better than the accuracy of either model individually.

Yoon et al. [30] proposed a CNN-BiLSTM architecture for predicting document-level sentiments with multi-channel embeddings using Word2vec word embedding. The models were applied on different datasets and obtained promising but modest performance of between 51.97% and 70.08%.

3.2. Document Level Sentiment Analysis

The classification of documents, although less popular than short text analysis of social networks, can be very useful in different areas, such as the analysis of political opinions [31–33] expressed in the press, the analysis of users' opinions [34] and press coverage. In our previous work [35], we compared the performance of CNN and LSTM models for long text and found that combining Doc2vec and CNN models slightly surpassed the performance of RNN. This is because CNN uses the Doc2Vec model, which detects the polarity of the entire document.

Missen et al. [36] addressed opinion analysis in documents using the "word to document level" approach using a corpus of documents. They first determined the polarity of words, then combined the polarity of the words in a sentence to determine the polarity of the sentence and finally combined the polarity of the sentences to calculate the final score to detect the polarity of the document.

Yessenalina et al. [37] proposed a joint two-level approach for document sentiment classification by automatically generating annotator rationales.

Rao et al. [38] modeled long texts to exploit semantic relationships between sentences in the sentiment classification at document level. They proposed SR-LSTM and SSR-LSTM, which are based on DeepRNN, with two layers. The first one represents the semantics of sentences, and the second one codes the sentence relationships in the representation of the document. Applied on the IMDB and yelp2015 databases, respectively, the SR-LSTM model obtained an accuracy of 44.0% and 63.9% and the SSR-LSTM model obtained an accuracy of 44.3% and 63.8%.

Fu et al. [39] proposed a new method called Bag Of Meta-Words. They represented the document with meta-word vectors, such that each vector designated semantic information of the document. The particularity of this method is that it captures the semantic meanings of the document. They obtained a 90.88% accuracy.

The related work is summarized in Table 1.

Table 1. Related works.

Word Embedding	Level	Model	Accurary
WORD2VEC	Sentence Level	CharSCNN [25]	86.4%
		BLSTM-2DPooling [26]	88.3%
		BLSTM-2DCNN [26]	89.5%
		CNN-Static [3]	89.6%
	Document Level	MutiChannel CNN-BiLSTM [30]	51.9–70.0%
GLOVE	Word Level	CNN-LSTM [28]	89.0%
	Sentence Level	CNN-LSTM [27]	89.9%
		CNN-BiLSTM [29]	89.7%
	Document Level	SR-LSTM [38]	44.0–63.9%
		SSR-LSTM [38]	44.3–63.8%
BOMW	Document Level	BOMW [39]	90.8%

4. Proposed Model: CNN-BiLSTM and Doc2vec for Document-Level Sentiment Analysis

This model combines two neural networks, namely CNN and BiLSTM. We performed this combination to test the adaptation of CNN with BiLSTM, since the latter is known for its performance in opinion analysis. The strong point of this model is that it allows extracting the maximum amount of information from documents using CNN convolution layers. This output becomes the BiLSTM input, which allows keeping the chronological order between the data in both directions.

4.1. Model Overview and Motivation

The combination of CNN and RNN models requires a particular design, since each model has a specific architecture and its own strengths:

- CNN is known for its ability to extract as many features as possible from the text.
- LSTM/BiLSTM keeps the chronological order between words in a document, thus it has the ability to ignore unnecessary words using the delete gate.

The purpose of combining these two models is to create a model that takes advantage of the strengths of CNN and BiLSTM, so that it captures the features extracted using CNN, and uses them as an LSTM input. Therefore, we develop a model that meets this objective, such that the vectors built in the word embedding part are used as convolutional neural network input. Then, four filters of sizes 2, 3, 4 and 5, respectively, are applied 100 times each. After each filter, a layer of max pooling is applied to update and reduce the size of the data.

Then, the results of all max pooling layers are concatenated to build the BiLSTM input, which applys a BiLSTM layer to filter the information, using its three gates. The output of this step is the input of the fully connected layer, which links each piece of input information with a piece of output information. Finally, we apply the softmax function as an activation function to assign classes to articles in order to produce the desired output.

Thus, we propose the following architecture composed of three parts, which are described in more detail below (Figure 2):

- **Pre-processing part**: In this stage, data cleansing and pre-processing are performed. Then, distributed document representation using Doc2Vec embedding is applied to prepare data for convolution. The resulting vector is passed as an input to the next stage.
- **Convolution part**: In this stage, convolution and max pooling layers are applied for feature extraction to extract high level features. The output of this stage is the input of the next stage.

- **BiLSTM/fully connected part**: In this stage, BiLSTM and fully connected layers are applied for document sentiment classification. The output of this stage is the final classification of the document (as positive, negative or neutral).

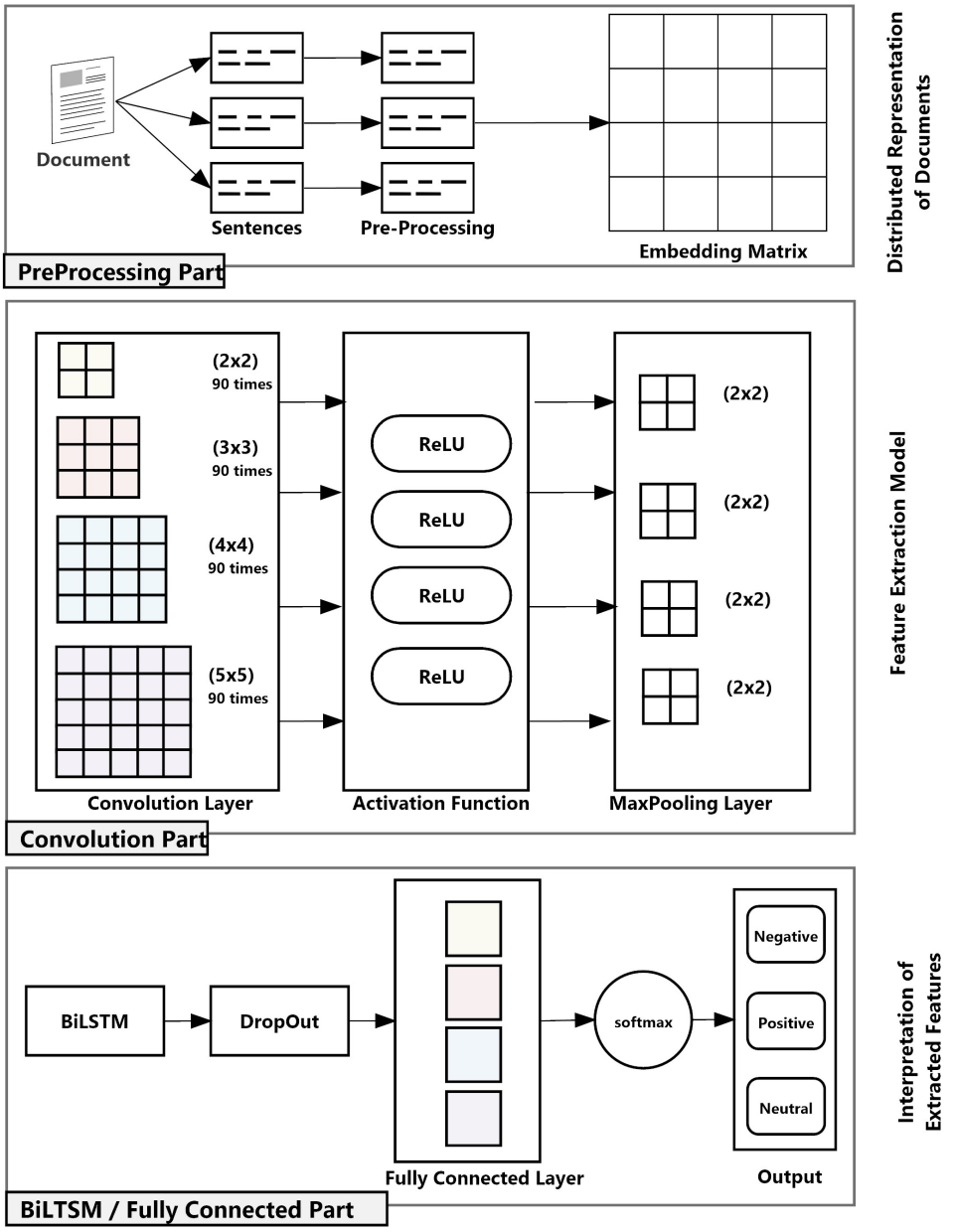

Figure 2. CNN-BiLSTM general architecture.

4.2. Document Representation

Word embedding allows preparing the data in the appropriate vector format for the input of artificial neural networks. Since we dealt with long text, we tested both Doc2vec and Word2vec with a large size to represent each word/paragraph in a document by a vector in order to build the entire word/document vector. Moreover, we combined the two Word2vec/Doc2vec models, namely CBOW/DBOW and Skip-gram/DM, to obtain better performance.

Word2vec creates from the words of the document the corresponding vectors. It is based on two approaches, namely CBOW and Skip-gram, which have been concatenated to improve performance.

It is widely used for opinion analysis in short texts that are specific to social networks, but it is also beginning to be used for processing long texts.

Doc2vec creates vectors for each sentence/paragraph of the document. It is also based on two techniques, namely DBOW, which is equivalent to CBOW from Word2vec, and DM, which is equivalent to Skip-gram. We concatenated these two techniques to achieve high accuracy. It is newer method than Word2vec, and it is specific to document processing.

We studied Doc2vec embeddings applied to large document classification, as Doc2vec has proven to outperform Word2vec robustly on different datasets [40]. The output is a matrix that represents the links among words/sentences/paragraphs using the word embedding representation, in such a way that each word/sentence/paragraph has an equivalent vector of fixed length in the matrix.

4.3. Convolution Layer

The convolution layer aims to explore the combination between the different sentences/paragraphs of the document, using filters of size t. A convolutional neural network (CNN) is a feature-extracting architecture and is meant to be integrated in a larger network [6].

An n-gram is a sub-sequence of n adjacent words built from a given sequence. The fundamental principle is to compute, from a given sequence of words, the likelihood function of the appearance of the next word defined as follows:

$$p(w_i|w_1, \ldots, w_{i-1}) = p(w_i|w_{i-(n-1)}, w_{i-(n-2)}, \ldots, w_{i-1})$$

In the convolution layer, filters serve as n-gram detectors; each filter searches for a specific class of n-grams and assigns them high scores. The detected grams with highest score pass the max pooling operation [41].

Four filters of size 2, 3, 4 and 5 are applied, 100 times each.

- The first one applies 90 bigrams filters of size 2.
- The second applies 90 trigram filters of size 3.
- The third one applies 90 four-gram filters of size 4.
- The fourth applies 90 five-gram filters of size 5.

After each filter, a layer of max pooling is applied to update and reduce the size of the data. Then, the results of all max pooling layers are concatenated to build the BiLSTM input.

4.4. Activation Layer

Each convolution layer applies the Rectified Linear Unit (ReLU) function, allowing each negative output to be replaced, which is treated as unnecessary information for the neural network, with a 0, hence reducing the non-linearity of the network.

4.5. Regularization

Regularization is managed through several functions that organize a complex neural network to avoid overfitting that impacts the performance of deep learning models.

We use the two main types, dropout [42] and L2, which consist in penalizing large weights in order to optimize the neural network.

4.6. Optimization

Optimization is used in training deep learning algorithms for updating the model parameters (weights and bias values) across iterations. There are different optimizations strategies that calculate appropriate and optimum values for these parameters such as Stochastic Gradient Descent (SGD) [43,44] or Adaptive Moment Estimation (Adam) [45].

SGD is the classical non-adaptive optimization algorithms used to optimize deep learning networks that use a single learning rate which does not change during training. Adam is an extension method to SGD that uses an adaptive learning rate to optimize the networks that converges very quickly and outperforms SGD [45].

4.7. BiLSTM Layer

This layer keeps the sequential order between the data. It allows detecting the links between the previous inputs and the output. The entry of this layer is the concatenation of the max pooling results.

5. Experimental Results

5.1. DataSet

We built a dataset containing 2003 French articles from national and international newspapers (TelQuel, Aujourd'hui, Le Figaro, and LeMonde, among others). Each article contains an average of 4000 words. The articles were scrapped online and then manually labeled to build the dataset. The distribution of polarity is as follows: 1247 neutral tone articles, 474 positive articles and 282 negative ones (Table 2).

Table 2. Dataset description.

Total	2003
Positive	474
Neutral	1247
Negative	282

The dataset with a total of 2003 entries was divided into three parts, namely training, validation and testing, distributed as follows: 75% (1502 entries) for model training and 25% (501 entries) distributed for validation (375 entries) and testing (126 entries).

The extracted articles relate to political affairs, opinions on a given company and general current events, with a predominance of political articles that express an opinion on a recent event.

Political opinion can be expressed on elections, parties, political representatives, and a given country. This information requires special processing, as most strategic decisions are based on it. In particular, political and current affairs information plays an active role in shaping public opinion because the majority of citizens consider it a priority source of information and give it great credibility.

5.2. Results

We compared the performance of different configurations for processing long-text documents. Thus, the accuracy was computed according to three different iterations (i.e., 6, 8 and 10), between two values of the batch_size (i.e., 32 and 64), and the optimizers SGD and Adam (Table 3, the highest value is highlighted in red).

Accuracy refers to the proportion of correct predictions made by the model.

$$\text{Accurary} = \frac{\text{Correct Predictions}}{\text{Total Predictions}}$$

Table 3. CNN-BiLSTM configurations.

Word Embedding	Regularizer	Epochs	Batch size	Optimizer	Accuracy
Doc2vec	L2	8	32	Adam	**90.66%**
				SGD	62.30%
			64	Adam	88.89%
				SGD	63.40%
		6	32	Adam	85.20%
				SGD	61,07%
			64	Adam	86.89%
				SGD	60.30%
		10	32	Adam	87.23%
				SGD	63.56%
			64	Adam	89.30%
				SGD	64.45%

The CNN-BiLSTM model achieved an accuracy of 90.66% (Figure 3).

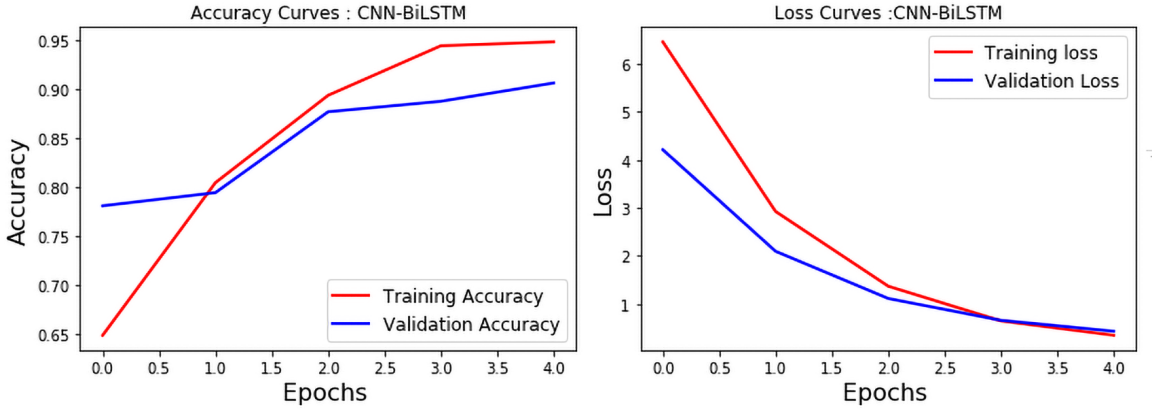

Figure 3. CNN-BiLSTM accuracy and loss.

We compared the performance of CNN-BiLSTM with Doc2vec word embedding with different configurations of CNN, LSTM, BiLSTM and CNN-LSTM. For all models, we measured the accuracy.

5.3. Comparaison

We compared the CNN-BiLSTM model with CNN, LSTM, BiLSTM and CNN-LSTM.

5.3.1. CNN Model

The CNN model is powerful in terms of feature extraction, which is very interesting in the opinion analysis, especially when dealing with long articles, and the extraction of its features remains difficult.

The CNN model was configured as follows:

- The maximum number of vectors that can be created from a document was 600.
- The size of the data of the input of the convolutional neural network was 600.
- The four convolution layers with Relu activation function were as follows:
  - The first one applies 90 bigrams filters of size 2.
  - The second one applies 90 trigram filters of size 3.
  - The third one applies 90 four-gram filters of size 4.
  - The fourth one applies 90 five-gram filters of size 5.

- The max pooling layer: We applied a max pooling layer after each convolution layer, and then concatenated the different max pooling layers to produce a single fixed size output.
- There was a fully connected layer.
- The softmax activation function allowed linking the obtained results with the appropriate class.
- The loss function is indispensable for compiling the "Sparse-categorical-crossentropy" model. We used the sparse one as there are three classes in integer and categorical format.
- The optimizer was "Adam".

The CNN model gave a high performance that reached 88.00% in terms of accuracy (Figure 4).

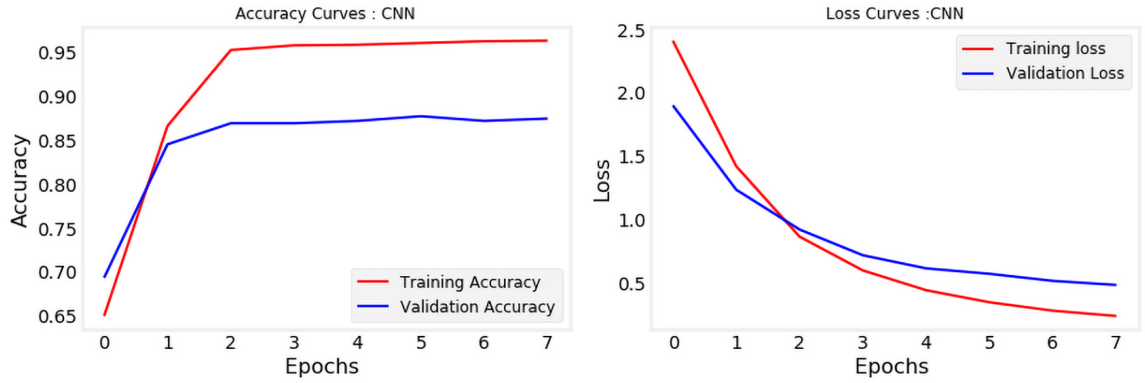

Figure 4. CNN accuracy and loss.

5.3.2. LSTM/BiLSTM Models

Recurrent neural networks, and specifically LSTM and BiLSTM, are known for their ability to keep the chronological order between data, which is very important when analyzing long-text opinions. LSTM/BiLSTM is different from CNN, thus it does not require a model to create input vectors.

The LSTM/BiLSTM models were configured as follows:

- The maximum size of the document vectors was 4676.
- There was a LSTM/BiLSTM layer.
- There was a fully connected layer.
- There was a softmax activation function.
- The loss function was "Sparse-categorical-crossentropy".
- The optimizer was "Adam".

The LSTM provided an accuracy of 85.87% (Figure 5) and the BiLSTM an accuracy of 86.40% (Figure 6). These models were close to CNN with Doc2vec.

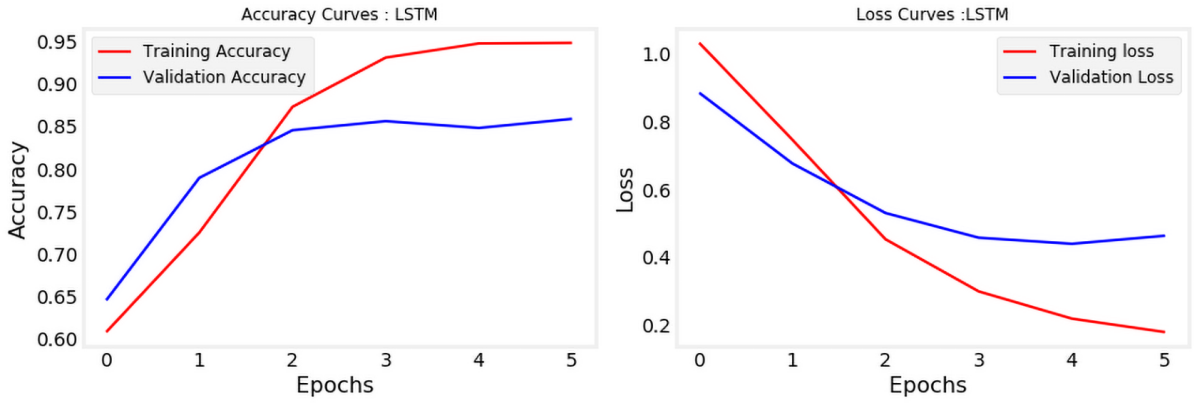

Figure 5. LSTM accuracy and loss.

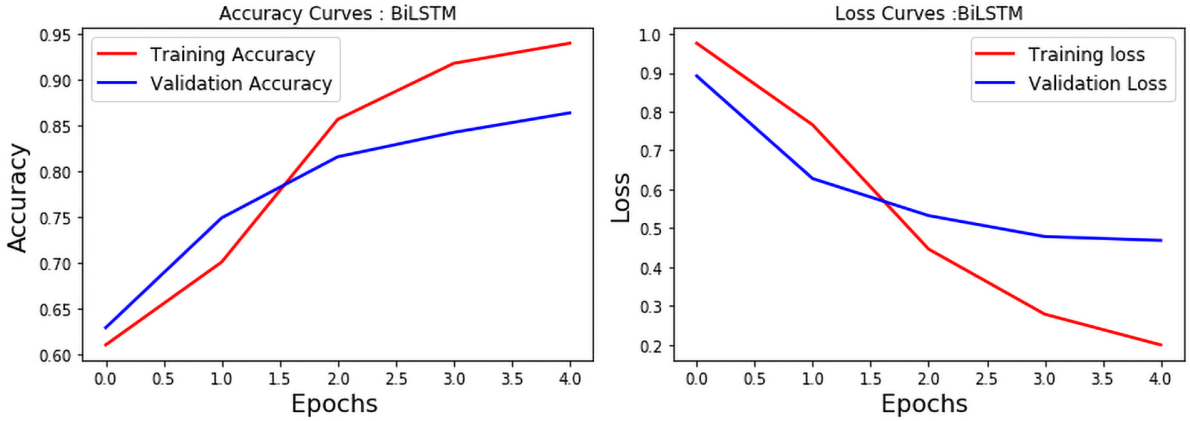

Figure 6. BiLSTM accuracy and loss.

5.3.3. CNN-LSTM Model

The CNN-LSTM model was configured as follows:

- The maximum number of vectors that can be created from a document was 600.
- The size of the data of the input of the convolutional neural network was 600.
- The four layers of convolution were as follows:
  - The first one applied 90 bigrams filters of size 2.
  - The second applied 90 trigram filters of size 3.
  - The third one applied 90 four-gram filters of size 4.
  - The fourth applied 90 five-gram filters of size 5.
- There was a max pooling layer. We applied a max pooling layer after each convolution layer, and then concatenated the different max pooling layers to produce a single fixed size output.
- There was a LSTM layer.
- There was a fully connected layer.
- The softmax activation function transformed the vector into probability to define the class of each output.
- The loss function, essential to compile the model, was "Sparse-categorical-crossentropy"
- The "Adam" optimizer was used.

This combination achieved an accuracy of 90.13% (Figure 7), which confirmed LSTM's added value compared to CNN.

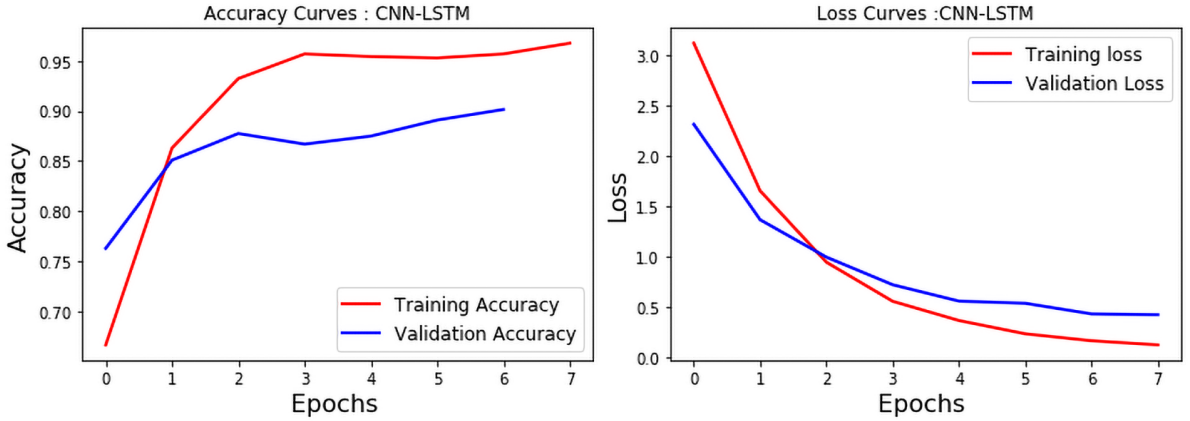

Figure 7. CNN-LSTM accuracy and loss.

5.4. Discussion

The comparison of the performance of different deep learning models confirmed the interest of the CNN-BiLSTM with Doc2vec, the pre-trained model of sentence/paragraph representation (Table 4, the highest value is highlighted in red).

Table 4. Model comparaison.

Word Embedding	Model	Accurary
Doc2vec	CNN	88.00%
	LSTM	85.87%
	BiLSTM	86.40%
	CNN-LSTM	90.13%
	CNN-BiLSTM	**90.66%**

Doc2vec performed better than Word2vec, given the context of long articles, as Word2vec is more appropriate for sentence-level short text analysis.

This result shows the effect of BiLSTM in keeping the link and order between the data in two directions in order to understand the context. In the CNN characterized by its ability to extract the features of the data, this combination allowed benefitting from the strengths of each model.

6. Conclusions

In this paper, we present a combination of convolutional and bidirectional recurrent neural networks for document-level sentiment analysis with Doc2vec Embedding. The combined CNN-BiLSTM model gives good results over the long text, since it benefits from the CNN's ability to extract features and the BiLSTM's characteristic to learn long-term bidirectional dependencies of the text. Furthermore, Doc2vec embedding processes the representation of the text at paragraph level, which is more suitable for the classification of a long-text document, unlike the traditional short text processing typical of social networks.

The model was trained on a dataset of French articles from online newspapers. For validation, we compared five deep learning models, namely CNN, LSTM, BiLSTM, CNN-LSTM and CNN-BiLSTM, with different configurations to conclude through experimental studies that the CNN-BiLSTM model is the one that achieved the best accuracy of 90.66%.

In future work, the proposed model can be adapted for Arabic documents [46], include attention mechanism for document-level sentiment analysis [47,48] and investigate other word embeddings such as Glove [49] and Fasttext [50].

Author Contributions: Conceptualization, M.R., M.M. and S.Y.; Data curation, S.B.; Methodology, M.R., M.M., S.Y. and S.B.; Software, S.B.; Validation, M.R.; Writing – original draft, M.R. and S.B.; Writing – review & editing, M.R. and M.M.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Liu, B. Sentiment analysis and opinion mining. *Synth. Lect. Hum. Lang. Technol.* **2012**, *5*, 1–167. [CrossRef]
2. Nasukawa, T.; Yi, J. Sentiment analysis: Capturing favorability using natural language processing. In Proceedings of the 2nd International Conference on Knowledge Capture, Austin, TX, USA, 4–6 December 2003; pp. 70–77.

3. Kim, Y. Convolutional Neural Networks for Sentence Classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*; Association for Computational Linguistics: Doha, Qatar, 2014; pp. 1746–1751.
4. Sundermeyer, M.; Schlüter, R.; Ney, H. LSTM neural networks for language modeling. In Proceedings of the Thirteenth annual conference of the international speech communication association, Portland, OR, USA, 9–13 September 2012.
5. Yin, W.; Kann, K.; Yu, M.; Schütze, H. Comparative study of CNN and RNN for natural language processing. *arXiv* **2017**, arXiv:1702.01923.
6. Goldberg, Y. Neural network methods for natural language processing. *Synth. Lect. Hum. Lang. Technol.* **2017**, *10*, 1–309. [CrossRef]
7. Pang, B.; Lee, L. Opinion mining and sentiment analysis. *Found. Trends Inf. Retr.* **2008**, *2*, 1–135. [CrossRef]
8. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed representations of words and phrases and their compositionality. In Proceedings of the Advances in Neural Information Processing Systems, Tahoe, NV, USA, 5–10 December 2013; pp. 3111–3119.
9. Le, Q.; Mikolov, T. Distributed representations of sentences and documents. In Proceedings of the 31th International Conference on Machine Learning, Beijing, China, 21–26 June 2014; pp. 1188–1196.
10. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
11. Gers, F. Long Short-Term Memory in Recurrent Neural Networks. Ph.D. Thesis, Leibniz Universitat Hannover, Hannover, Germany, 2001.
12. Greff, K.; Srivastava, R.K.; Koutník, J.; Steunebrink, B.R.; Schmidhuber, J. LSTM: A search space odyssey. *arXiv* **2015**, arXiv:1503.040692.
13. Lai, S.; Xu, L.; Liu, K.; Zhao, J. Recurrent convolutional neural networks for text classification. In Proceedings of the 29th AAAI Conference on Artificial Intelligence, Austin, TX, USA, 25–30 January 2015.
14. Schuster, M.; Paliwal, K.K. Bidirectional recurrent neural networks. *IEEE Trans. Signal Process.* **1997**, *45*, 2673–2681. [CrossRef]
15. Tai, K.S.; Socher, R.; Manning, C.D. Improved semantic representations from tree-structured long short-term memory networks. *arXiv* **2015**, arXiv:1503.00075.
16. Gamal, D.; Alfonse, M.; M El-Horbaty, E.S.; M Salem, A.B. Analysis of Machine Learning Algorithms for Opinion Mining in Different Domains. *Mach. Learn. Knowl. Extr.* **2019**, *1*, 224–234. [CrossRef]
17. Zhang, L.; Wang, S.; Liu, B. Deep learning for sentiment analysis: A survey. *arXiv* **2018**, arXiv:1801.07883.
18. Go, A.; Bhayani, R.; Huang, L. *Twitter Sentiment Classification Using Distant Supervision*; CS224N Project Report; Stanford University: Stanford, CA, USA, 2009.
19. Dong, L.; Wei, F.; Tan, C.; Tang, D.; Zhou, M.; Xu, K. Adaptive recursive neural network for target-dependent twitter sentiment classification. In Proceedings of the 52nd annual meeting of the association for computational linguistics (volume 2: Short papers), Baltimore, MD, USA, 22–27 June 2014; Volume 2, pp. 49–54.
20. Tang, D.; Wei, F.; Qin, B.; Liu, T.; Zhou, M. Coooolll: A deep learning system for twitter sentiment classification. In Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014), Dublin, Ireland, 23–24 August 2014; pp. 208–212.
21. Severyn, A.; Moschitti, A. Twitter sentiment analysis with deep convolutional neural networks. In Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, Santiago, Chile, 9–13 August 2015; pp. 959–962.
22. Jianqiang, Z.; Xiaolin, G.; Xuejun, Z. Deep convolution neural networks for Twitter sentiment analysis. *IEEE Access* **2018**, *6*, 23253–23260. [CrossRef]
23. Uysal, A.K.; Gunal, S. The impact of preprocessing on text classification. *Inf. Process. Manag.* **2014**, *50*, 104–112. [CrossRef]
24. Petz, G.; Karpowicz, M.; Fürschuß, H.; Auinger, A.; Winkler, S.M.; Schaller, S.; Holzinger, A. On text preprocessing for opinion mining outside of laboratory environments. In *International Conference on Active Media Technology*; Springer: Berlin, Germany, 2012; pp. 618–629.
25. Dos Santos, C.; Gatti, M. Deep convolutional neural networks for sentiment analysis of short texts. In Proceedings of the COLING 2014, the 25th International Conference on Computational Linguistics, Dublin, Ireland, 23–29 August 2014; pp. 69–78.

26. Zhou, P.; Qi, Z.; Zheng, S.; Xu, J.; Bao, H.; Xu, B. Text classification improved by integrating bidirectional LSTM with two-dimensional max pooling. *arXiv* **2016**, arXiv:1611.06639.
27. Wang, X.; Jiang, W.; Luo, Z. Combination of convolutional and recurrent neural network for sentiment analysis of short texts. In Proceedings of the COLING 2016, the 26th International Conference on Computational Linguistics, Osaka, Japan, 11–16 December 2016; pp. 2428–2437.
28. Yenter, A.; Verma, A. Deep CNN-LSTM with combined kernels from multiple branches for IMDb review sentiment analysis. In Proceedings of the 2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON), New York, NY, USA, 19–21 October 2017; pp. 540–546.
29. Shen, Q.; Wang, Z.; Sun, Y. Sentiment analysis of movie reviews based on cnn-blstm. In *International Conference on Intelligence Science*; Springer: Berlin, Germany, 2017, pp. 164–171.
30. Yoon, J.; Kim, H. Multi-Channel Lexicon Integrated CNN-BiLSTM Models for Sentiment Analysis. In Proceedings of the 29th Conference on Computational Linguistics and Speech Processing (ROCLING 2017), Taipei, Taiwan, 27–28 November 2017; pp. 244–253.
31. Sobkowicz, P.; Kaschesky, M.; Bouchard, G. Opinion mining in social media: Modeling, simulating, and forecasting political opinions in the web. *Gov. Inf. Q.* **2012**, *29*, 470–479. [CrossRef]
32. Sarmento, L.; Carvalho, P.; Silva, M.J.; De Oliveira, E. Automatic creation of a reference corpus for political opinion mining in user-generated content. In *Proceedings of the 1st International CIKM Workshop on Topic-Sentiment Analysis for Mass Opinion*; ACM: New York, NY, USA, 2009; pp. 29–36.
33. Durant, K.T.; Smith, M.D. Mining sentiment classification from political web logs. In Proceedings of Workshop on Web Mining and Web Usage Analysis of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (WebKDD-2006), Philadelphia, PA, USA, 20–23 August 2006.
34. Hu, Y.H.; Chen, Y.L.; Chou, H.L. Opinion mining from online hotel reviews–A text summarization approach. *Inf. Process. Manag.* **2017**, *53*, 436–449. [CrossRef]
35. Yousfi, S.; Rhanoui, M.; Mikram, M. Comparative Study of CNN and RNN For Opinion Mining in Long Text. In Proceeding of the International Conference on Modern Intelligent Systems Concepts, Rabat, Morocco, 12–13 December 2018.
36. Missen, M.M.S.; Boughanem, M.; Cabanac, G. Opinion mining: reviewed from word to document level. *Soc. Netw. Anal. Min.* **2013**, *3*, 107–125. [CrossRef]
37. Yessenalina, A.; Yue, Y.; Cardie, C. Multi-level structured models for document-level sentiment classification. In Proceedings of the 2010 conference on empirical methods in natural language processing. Association for Computational Linguistics, Cambridge, MA, USA, 9–11 October 2010; pp. 1046–1056.
38. Rao, G.; Huang, W.; Feng, Z.; Cong, Q. LSTM with sentence representations for document-level sentiment classification. *Neurocomputing* **2018**, *308*, 49–57. [CrossRef]
39. Fu, M.; Qu, H.; Huang, L.; Lu, L. Bag of meta-words: A novel method to represent document for the sentiment classification. *Expert Syst. Appl.* **2018**, *113*, 33–43. [CrossRef]
40. Lau, J.H.; Baldwin, T. An empirical evaluation of doc2vec with practical insights into document embedding generation. *arXiv* **2016**, arXiv:1607.05368.
41. Jacovi, A.; Shalom, O.S.; Goldberg, Y. Understanding convolutional neural networks for text classification. *arXiv* **2018**, arXiv:1809.08037.
42. Hinton, G.E.; Srivastava, N.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R.R. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv* **2012**, arXiv:1207.0580.
43. LeCun, Y.A.; Bottou, L.; Orr, G.B.; Müller, K.R. Efficient backprop. In *Neural Networks: Tricks of the Trade*; Springer: Berlin, Germany, 2012; pp. 9–48.
44. Bottou, L. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*; Springer: Berlin, Germany, 2010; pp. 177–186.
45. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
46. Shoukry, A.; Rafea, A. Sentence-level Arabic sentiment analysis. In Proceedings of the 2012 International Conference on Collaboration Technologies and Systems (CTS), Denver, CO, USA, 21–25 May 2012; pp. 546–550.
47. 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, TA, USA, 1–5 November 2016; pp. 606–615.

48. 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.
49. Pennington, J.; Socher, R.; Manning, C. Glove: Global vectors for word representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1532–1543.
50. Bojanowski, P.; Grave, E.; Joulin, A.; Mikolov, T. Enriching word vectors with subword information. *Trans. Assoc. Comput. Linguist.* **2017**, *5*, 135–146. [CrossRef]