applied sciences

MDPI

Article

Senti-Sequence: Learning to Represent Texts for Sentiment Polarity Classification

Andres Ramos Magna [1], Juan Zamora [2,*] and Hector Allende-Cid [3,4,5]

[1] Departamento de Tecnologías de Información, Universidad de Valparaíso, Calle Prat 856, Valparaíso 2361864, Chile; andres.ramos@uv.cl
[2] Instituto de Estadística, Pontificia Universidad Católica de Valparaíso, Avenida Brasil 2830, Valparaíso 2340025, Chile
[3] Escuela de Ingeniería Informática, Pontificia Universidad Católica de Valparaíso, Avenida Brasil 2241, Valparaíso 2362807, Chile; hector.allende@pucv.cl or hector.allende-cid@iais.fraunhofer.de
[4] Knowledge Discovery, Fraunhofer-Institute of Intelligent Analysis and Information Systems (IAIS), Schloss Birlinghoven, 1, 53757 Sankt Augustin, Germany
[5] Lamarr Institute for Machine Learning and Artificial Intelligence, 44227 Dortmund, Germany
* Correspondence: juan.zamora@pucv.cl; Tel.: +56-322274058

Abstract: The sentiment analysis task seeks to categorize opinionated documents as having overall positive or negative opinions. This task is very important to understand unstructured text content generated by users in different domains, such as online and entertainment platforms and social networks. In this paper, we propose a novel method for predicting the overall polarity in texts. First, a new polarity-aware vector representation is automatically built for each document. Then, a bidirectional recurrent neural architecture is designed to identify the emerging polarity. The attained results outperform all of the algorithms found in the literature in the binary polarity classification task.

Keywords: sentiment analysis; machine learning; deep learning

check for updates

Citation: Ramos Magna, A.; Zamora, J.; Allende-Cid, H. Senti-Sequence: Learning to Represent Texts for Sentiment Polarity Classification. *Appl. Sci.* **2024**, *14*, 1033. https://doi.org/10.3390/app14031033

Academic Editor: Giacomo Fiumara

Received: 6 December 2023
Revised: 6 January 2024
Accepted: 16 January 2024
Published: 25 January 2024

1. Introduction

Sentiment analysis seeks to identify the polarity of a document as positive or negative [1]. Formally, it is the computational study of emotions, feelings, comments, and opinions described in a text [2], which has been applied in multiple classification tasks in business areas [3,4], medical [5], and social networks [6]. This task is extremely important for understanding user-generated text content in different domains, such as online shopping, entertainment platforms, and social networks. Machine learning techniques such as support vector machine (SVM) and naive Bayes have been successfully employed in feature-based approaches [7,8]. Despite the important progress brought by these methods, the increasing complexity of the classification task given by the variable length and hazy emotions associated with the text content pose tough challenges to every handcrafted feature-based approach. More recently, deep recurrent neural networks have been successfully employed for this task, attaining the highest performance [9–11].

Today, sentiment analysis is a growing research field influenced by multiple research fields, such as text mining, machine learning, computational linguistics, and natural language processing [12]. This process consists of classifying the content of the text based on a possible expression of opinion or sentiment of the writer, trying to identify whether the expression is positive, negative, or neutral and, in many cases, boiling down to positive and negative.

To carry out this task, multiple techniques are proposed to identify the sentiment associated with the texts with a certain degree of certainty. To classify the documents, techniques are used from two main approaches: those based on machine learning and those based on sentilexicons [13]. Additionally, it is possible to distinguish three main categories in sentiment classification [14] in the literature: document-level, sentence-level, and aspect-level. The category consists of determining the polarity underlying the content of a text document. In the second category, each sentence within a document is classified as positive, negative, or neutral. Finally, the last category determines the opinions expressed on different entity features.

In this work, we propose a document-level approach for binary sentiment classification. The contribution is twofold. First, we develop a method that learns to classify words based on multiple sentilexicon dictionaries. Second, we design a deep neural network architecture that captures the polarity oscillations marked by the sequence of terms within each document. The work is structured as follows. In Section 2, we present related works that deal with binary sentiment analysis and its discussion. In Section 3, we present our proposed method. Next, the experimental validation of the proposal is shown. Finally, we conclude with final remarks and future work.

2. Related Work

Document-level sentiment analysis approaches categorize opinionated documents as having overall positive or negative opinions [15]. Initially, the traditional bag-of-features approach to sentiment classification was employed. Automatic classifiers such as SVM, naive Bayes, and maximum entropy were used mostly on documents represented with n-grams or manually designed features [7,8,16–18]. Next, effort was put into the design of better word features that exploit language knowledge, and based on the work of Mikolov et al. [19], new dense word representations were built for sentiment classification [20–23].

Recently, deep recurrent neural networks have been employed for sentiment classification tasks [9–11,23,24]. Despite the long training times that these models present, one of the greatest advantages they offer is the capacity to automatically learn feature representations directly from the input word sequences. A particularly important approach to this work is that the polarities of these words are used to improve the solution quality in the classification task [25–28]. Often, the sources of this emotional information attached to each word are sentiment lexicons such as WordNet-Affect [29] and SentiWordNet [30].

In the same direction as our proposal, other approaches using recurrent neural networks with sentiment lexicon information have been successfully applied [10,23,31–34].

Shin et al. [31] integrated lexicon embeddings and an attention mechanism into convolutional neural networks. They concatenated scores coming from six different lexicons to build a sentiment word embedding. Then, these embeddings were padded and passed together with the word embeddings to the net. Experiments over movie reviews and tweets show a notorious improvement in accuracy and F1 measures when comparing the performances attained with and without lexicon embeddings.

Zeyang et al. [32] proposed a procedure for learning hidden representations of context and sentiment words using single-head and multihead attention weights jointly learned with lexicon and text document inputs.

Xiong et al. [23] considered that not all words within a tweet message have the same sentiment polarity as that of the whole document. Then, they learned sentiment-specific word embeddings by exploiting both a lexicon resource and distant supervised information and feeding several neural networks with a word representation combining lexicon and tweet sentiment information to obtain multilevel sentiment-enriched word embeddings.

Wu et al. [10] represented each sentence as a sequence of word vectors. Then, they proposed using sentiment lexicons to enhance the learning of word attention weights to classify sentences according to their polarity. Additionally, a hierarchical network architecture is presented to automatically build document representations that capture the contextual importance of each sentence.

Li et al. [33] exploited the emotional resources for the document sentiment classification task, the sentiment variations that the words present in different document locations, and addressed the information loss that occurs when word sequences are encoded into fixed-length vectors. To do this, they first proposed a bidirectional LSTM architecture (SAMF-

BiLSTM) that learns dense representations for words in a sentence by combining lexical and syntactic features to classify the sentiment underlying the sentence. The document-level classification is performed by a second model (SAMF-BiLSTM-D) that addresses this task as a sentence-sequence classification where the SAMF-BiLSTM model estimates the polarity of each sentence in a document.

More recently, Li et al. [34] mined the sentiment information from user reviews using a mixed convolutional and LSTM architecture enhanced with sentilexicon information.

Previous contributions in the document-level sentiment classification task have brought interesting ideas regarding the integration of the word lexicon information into deep learning models. The construction of sentiment-aware embeddings to integrating polarity scores from several sources has been considered. Nevertheless, two less-addressed issues remain whose resolution could improve classification accuracy: the lack of word sentiment when no polarity information is available and, more importantly, the exploitation of the relationship between the fluctuation of polarity along the documents and their target sentiments.

3. The Proposed Classification Framework

The overall idea of this proposal resides in creating a new representation of a document that portrays the fluctuation of the emotional charge through its text using a sequence of automatically identified positive and negative polarities. We address the document classification task by following a statement made by Xiong et al. [23] in which they pose that not all words within a text have the same sentiment polarity as that of the whole document.

The process consists of the three stages depicted in Figure 1. The first corresponds to creating a word polarity classifier based on multiple sentilexicons and dense word representations. In this manner, the classifier is trained to identify positive and negative words from their word embeddings and polarities extracted from several lexicons. In the second stage, each document is represented by a binary sequence denoting the polarities of all its constituent words. In the third stage, the deep sentiment classifier is trained. To better describe the complete process, each phase is presented below.

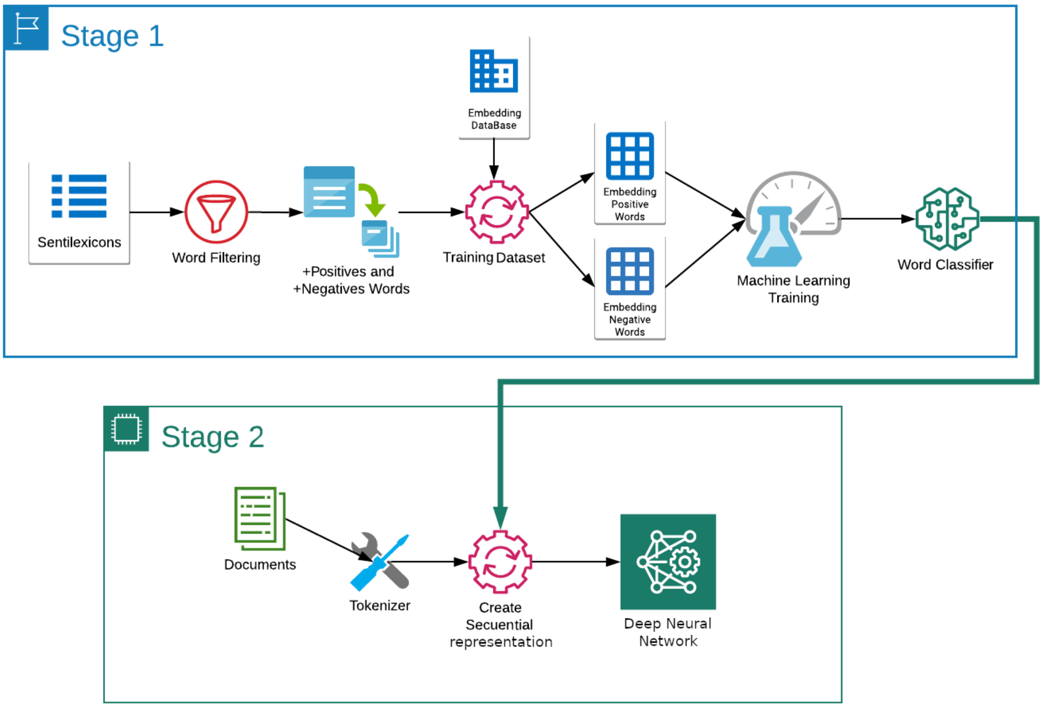


Figure 1. Overall view of the proposed method.

3.1. Automatic Identification of Word Polarities

Instead of building sentiment-aware embeddings (see [35,36] for some interesting approaches), we follow a simple yet effective strategy that consists of using several sentilexicons and word embeddings to train a vector classifier capable of predicting word polarity from its meaning vector. The overall process is depicted in the first stage of Figure 1.

To generate the ground truth information to train the classifier, the vector of each word in an embedding database is paired with its corresponding polarity, and the overall result is aggregated into a single dataset. To increase coverage of the vocabulary used by the method, the polarities were extracted from the following lexicons: SentiwordNet [37], SenticNet [38], SentiWords [39], and NLTK sentiment package [40] and positive and negative words [41].

After the training set is built, each word w will have a polarity score $\phi(w) \in \{-1, +1\}$ along with its embedding vector. Then, these tuples are fed into a binary classifier following a standard training procedure. Once the classifier is trained, new words, along with an available embedding, can be classified as positive or negative. The example shown in Figure 2 depicts the text processing tasks performed to generate a polarity vector from text content. This vector contains has values -1 and $+1$ for words whose polarity is predicted as negative and positive by the classifier respectively. A 0 value is used for words whose embedding is not available.

In this part of the process, any word embedding database and vector classifier capable of handling high-dimensional data can be employed. In the experimental section, we assess the performance of several alternatives for these two components and describe in detail those we chose.

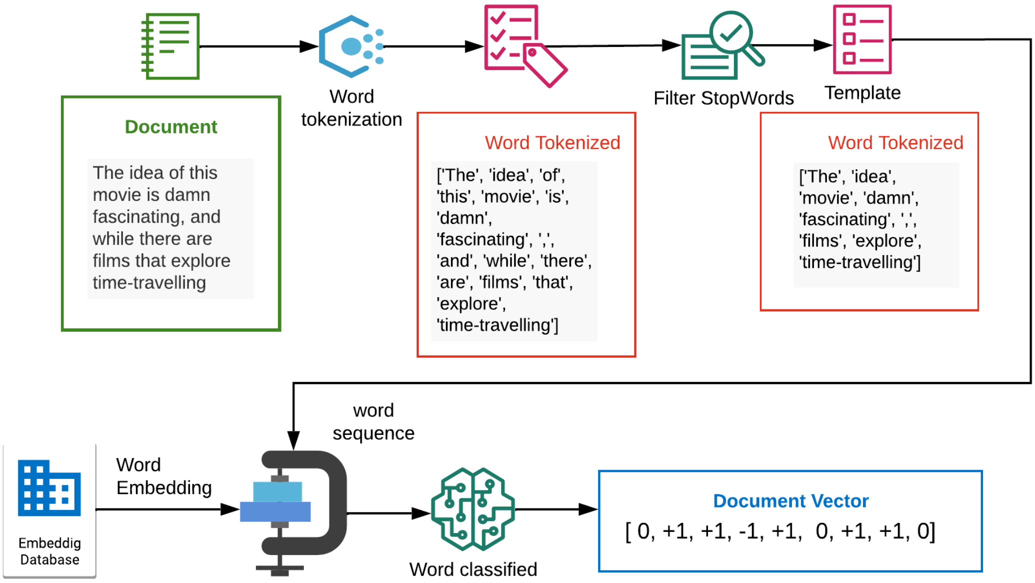

Word tokenization
Filter StopWords
Template
Document
The idea of this movie is damn fascinating, and while there are films that explore time-travelling
Word Tokenized
['The', 'idea', 'of', 'this', 'movie', 'is', 'damn', 'fascinating', ',', 'and', 'while', 'there', 'are', 'films', 'that', 'explore', 'time-travelling']
Word Tokenized
['The', 'idea', 'movie', 'damn', 'fascinating', ',', 'films', 'explore', 'time-travelling']
word sequence
Word Embedding
Embeddig Database
Word classified
Document Vector
[0, +1, +1, -1, +1, 0, +1, +1, 0]

Figure 2. Example of the document sequence generation procedure from its word polarities.

3.2. Bidirectional LSTM for Document Polarity Identification

The aim of this task consists of obtaining a single representative polarity for the whole document. To accomplish this, first, the polarity distribution within a text is represented as a sequence of marks denoting the sentiment of each word. These marks result from the polarity prediction task performed by the classifier proposed in Section 3.1. Then, an ad hoc classifier capable of dealing with these sequential data is employed to obtain the overall document polarity. The complete process is depicted in the second stage of Figure 1. As a special case, when no embedding is available for a given word, its mark in the sequence is a 0.

Without loss of generality, let us consider that there exists an embedding vector for each word present in the underlying text corpus. Then, each input document t containing a sequence of n_t words $[w_1, w_2, \ldots, w_{n_t}]$ is transformed into a binary vector

$[\phi(w_1), \phi(w_2), \ldots, \phi(w_{n_t})]$ by the trained polarity classifier, where each $\phi(w_i) \in \{-1, +1\}$ is defined in Section 3.1. The main idea behind our proposal is to build a dense vector representation that relates the polarity value of each input word with the polarity values of all its surrounding words within the document. In order to accomplish this task, we make use of the bidirectional recurrent network learning formulation. This is, to estimate a function G that takes $\phi(w_i)$ and outputs real values

$$\overrightarrow{h_i} = G(\phi(w_i), \overrightarrow{h}_{i-1})$$
$$\overleftarrow{h_i} = G(\phi(w_i), \overleftarrow{h}_{i+1})$$

to account for the nonlinear influence of polarities of words appearing before w_i and after in the text, respectively. The effect of this procedure is to map each binary value $\phi(w_i)$ onto a continuous polarity value that also considers the word context represented by its neighborhood within the text.

In the end, we need to estimate another function f that merges these two output values and predicts the overall text polarity.

These networks have long been used to model sequential data in natural language processing tasks, and despite their wide usage, they present problems with unstable gradients and very limited short memory. To overcome the last issue, long short-term memory (LSTM) is employed due to its capability of selecting, preserving, and extracting important inputs whenever needed. Bidirectional architectures have also been successfully employed for similar tasks in recent works [33,42].

Regarding some practical issues, a zero padding strategy is used to address the variable length of text documents. Additionally, each word polarity is processed by the network in order of appearance in the text and fed to the network. The underlying idea of providing contextual information for each word polarity is mainly carried by the long-term states of the two opposite LSTM layers. Figure 3 depicts the proposed architecture.

Furthermore, we pose that the binary valued polarity of each word is enriched by adding information of the entire text. Additionally, it is important to note that we use several recurrent units; hence, each binary polarity value is mapped onto several dimensions.

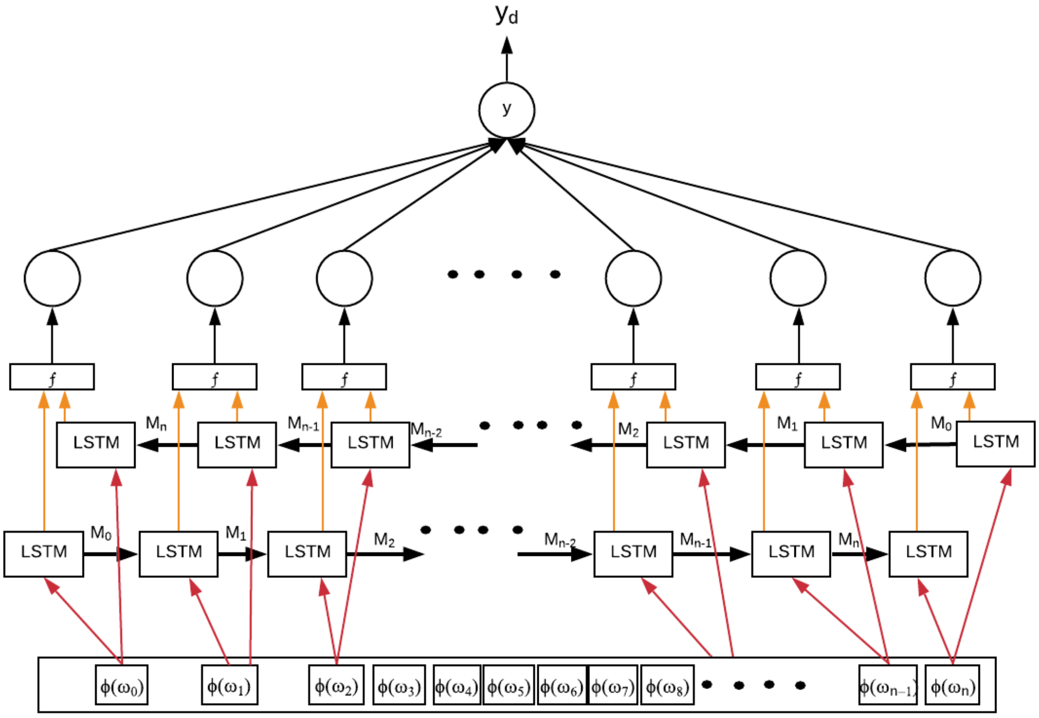


Figure 3. Bidirectional LSTM Architecture.

4. Experiments

4.1. Word Polarity Classifier Implementation

A key component of the architecture shown at the end of Stage 1 in Figure 1 corresponds to the classifier that automatically identifies the polarity of a word from its vector embedding as described in Section 3.1. To identify the best combination of pretrained word embeddings and vector classifiers, we assess the performance of five widely used machine learning methods along with three popular pretrained word embeddings.

Initially, the sentilexicons listed in Section 3.1 are merged. After we build the vocabulary and identify the word polarities from the sentilexicons, we assess the performance of each combination of vector classifier and pretrained word embedding by following a 10-fold cross-validation strategy. The comparison results obtained for each combination are depicted in Table 1.

Table 1. F1 sentilexicon results.

ML	ConceptNet [43]	Word2vec [44]	BERT [45]
Naive Bayes	**0.875**	0.741	0.706
Tree	**0.813**	0.632	0.691
SVM	**0.957**	0.811	0.816
Random Forest	**0.738**	0.672	0.707
KNN	**0.899**	0.755	0.738

Based on these results, we observe that ConceptNet consistently yields better classification results compared to word2vec or BERT. Additionally, the classifier that attains the best results along with ConceptNet is the SVM with an F1 value of 0.957. Thus, this tuple is used as the word polarity module employed in the subsequent experiments.

4.2. Experimental Setup

All experiments were carried on a an Intel® Core$^{\text{TM}}$ i7-7700K CPU (Intel) running at 4.20 GHz with a Nvidia Tesla K80 GPU (Nvidia). All developed models were built using PyTorch 1.7.1. To carry out the document classification process, we applied a standard five-fold cross-validation procedure, and each of the cross-validation executions was considered an independent result. We repeated each cross-validation process 10 times, and the final result reported for each model was obtained from the average performance over the test fold. The performance metric employed to select models is F1 measure due to its robustness under scenarios with imbalanced data. Additionally, accuracy, precision, and recall are reported.

4.3. Datasets

As show in Table 2, three benchmark collections are employed for the experimental validation of the proposal: A dataset of tweets that were taken from the popular Kaggle dataset Twitter US Airline, English movie reviews from IMDB, and Amazon Books reviews. The rationale of this dataset selection is to include very short and noisy texts such as tweets, mixed length and less noisy texts in IMDB, and finally, longer texts reviewing Amazon Books purchases. Stopwords were removed, and the maximum length for the the Twitter dataset is set to 40 words; for the IMDB dataset, 1574 words; and for the Amazon Books dataset, 4883 words.

Table 2. Size input vector in every case.

Database	Documents	Mean Vector Size
Twitter	2100	40
IMDB	6000	1574
Books dataset	2000	4883

4.4. Utility of the Recurrent and Bidirectional Characteristics

In order to assess the utility of the recurrent approach and the bidirectional recurrent architecture, two baseline neural network models are presented. First, a feedforward network with a single output neuron denoted as DENSE, and second, a unidirectional LSTM neural network with a fully connected layer denoted as LSTM + DENSE. The inputs of DENSE and LSTM + DENSE were the same as those that Senti-Sequence used for the classification process. The model selection and validation procedures applied are identical to those described in Section 4.2.

5. Results

Tables 3–6 show the results attained by the three proposed architectures. The recurrent and bidirectional method (Senti-Sequence) consistently attains the highest scores across all the datasets. Furthermore, the amount of dispersion relative to the average value is also higher for the baseline methods over the Twitter and Books datasets. This is, Senti-Sequence attained a coefficient of variation in the range of $[2\%, 7\%]$ in contrast with the range of $[2\%, 13\%]$ for the other two models across all metrics.

Table 3. F1 value attained by each method over all datasets. The bold numbers indicate the best values attained over each dataset.

Method	IMDB	Twitter	Books Generic
DENSE	0.975 ± 0.029	0.954 ± 0.052	0.838 ± 0.056
LSTM + Dense	0.827 ± 0.017	0.943 ± 0.051	0.677 ± 0.090
BILSTM (Senti-Sequence)	**0.990** ± 0.024	**0.963** ± 0.067	**0.990** ± 0.031

Table 4. Accuracy value attained by each method over all datasets. The bold numbers indicate the best values attained over each dataset.

Method	IMDB	Twitter	Books Generic
DENSE	0.975 ± 0.029	0.954 ± 0.052	0.823 ± 0.049
LSTM + Dense	0.824 ± 0.017	0.943 ± 0.051	0.962 ± 0.069
BILSTM (Senti-Sequence)	**0.989** ± 0.024	**0.963** ± 0.068	**0.990** ± 0.030

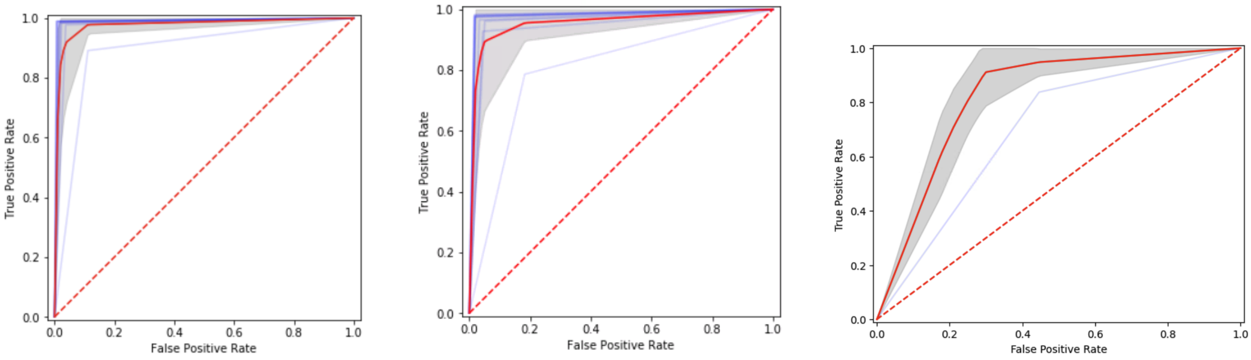
Table 5. Precision value attained by each method over all datasets. The bold numbers indicate the best values attained over each dataset.

Method	IMDB	Twitter	Books Generic
DENSE	0.975 ± 0.029	0.954 ± 0.052	0.852 ± 0.036
LSTM + Dense	0.831 ± 0.015	0.943 ± 0.051	0.678 ± 0.062
BILSTM (Senti-Sequence)	**0.990** ± 0.024	**0.963** ± 0.068	**0.990** ± 0.030

Table 6. Recall value attained by each method over all datasets. The bold numbers indicate the best values attained over each dataset.

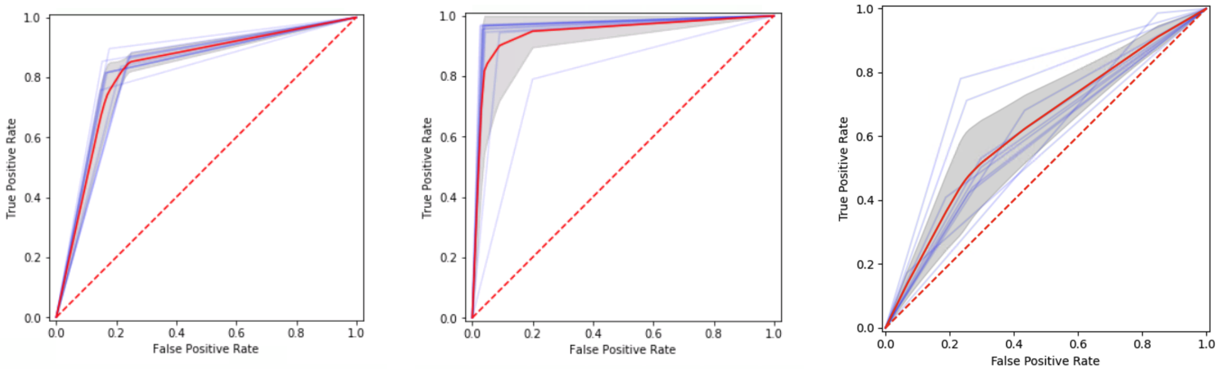
Method	IMDB	Twitter	Books Generic
DENSE	0.975 ± 0.029	0.954 ± 0.052	0.824 ± 0.049
LSTM + Dense	0.824 ± 0.016	0.943 ± 0.051	0.677 ± 0.062
BILSTM (Senti-Sequence)	**0.990** ± 0.024	**0.963** ± 0.068	**0.990** ± 0.030

In Figures 4–6, ROC curves for the performance attained by all the methods under evaluation across 10 runs are shown. Additionally, in each figure, a darker blue line denotes the maximum performance, the red line is the average, the lowest light-blue line is the minimum, and the shaded area represents the average curve $\pm\sigma$ (one standard deviation). In terms of the average performance classifier over the Twitter dataset, the three methods achieve comparable results. Nevertheless, when observing the average curves for the other two datasets, the pattern abruptly changes. That is, Senti-Sequence maintains a similar behavior with an average performance quite close to the perfect classification, but the other two methods approach closely to the nondiscrimination line in all their curves. This is particularly noticeable for the Books dataset in Figures 4c and 5c.



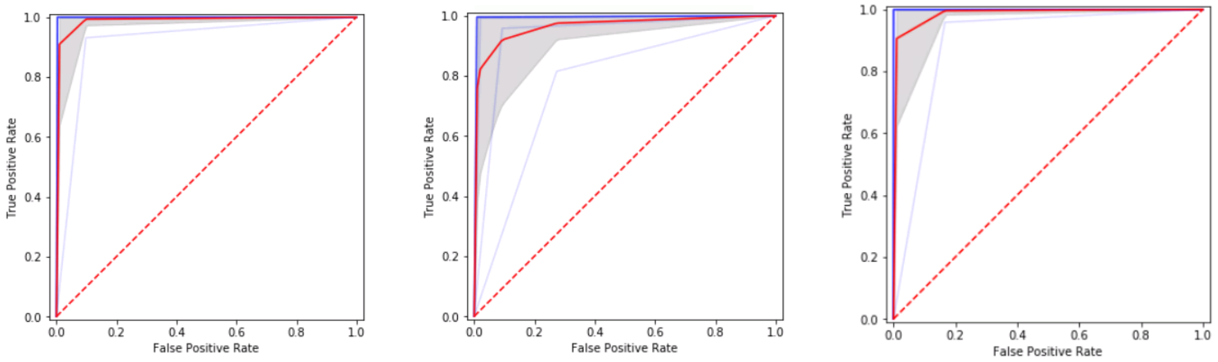
(**a**) (**b**) (**c**)

Figure 4. ROC curves for 10 runs of the DENSE model over each dataset. Curves in blue denote the maximum performance, red denotes the average, and dotted blue denotes the minimum performance. (**a**) IMDB dataset. (**b**) Twitter dataset. (**c**) Books generic dataset.



(**a**) (**b**) (**c**)

Figure 5. ROC curves for 10 runs of the LSTM + Dense model over each dataset. Curves in blue denote the maximum performance, red denotes the average, and dotted blue denotes the minimum performance. (**a**) IMDB dataset. (**b**) Twitter dataset. (**c**) Books generic dataset.



(**a**) (**b**) (**c**)

Figure 6. ROC curves for 10 runs of Senti-Sequence over each dataset. Curves in blue denote the maximum performance, red denotes the average, and dotted blue denotes the minimum performance. (**a**) IMDB dataset. (**b**) Twitter dataset. (**c**) Books generic dataset.

In Table 7, we observe the proposed method compared with recent works in the subject over the IMDB, Twitter, and Books generic datasets. We can see that the proposed model outperforms different types of models, including models that were state of the art in this classification.

Table 7. Comparison of accuracy results in multiple papers with IMDB, Twitter, and Books generic datasets. The bold numbers indicate the best values attained over each dataset.

Model	IMDB	Twitter	Books Generic
NB-weighted-BON + dv-cosine [11]	0.974	0.730	0.481
GraphStar [46]	0.960	0.960	0.650
L MIXED [47]	0.956	0.896	0.810
XLNet [48]	0.924	0.834	0.839
Senti-Sequence	**0.990**	**0.963**	**0.990**

6. Discussion

The three datasets employed differ in the number of average words per document and the number of documents. Books and IMDB datasets contain much larger documents in comparison to the Twitter dataset, as depicted in Table 2. All methods under evaluation decrease their performance on the Books dataset. This could be explained by the less amount of training data available plus the larger extension of the documents which also shows a greater variation in the polarity of the contained words. Nevertheless, it seems interesting that Senti-Sequence did not follow this pattern. In fact, the dataset having shorter documents presented more challenges to the correct classification of the method. However, the proposal attained the best F1 score. This suggests that Senti-Sequence is more sensitive to very short texts and also that it does not need very large collections.

It is noteworthy that in the case where the documents were larger, as in IMDB and Books, Senti-Sequence outperforms the other models. An identical behavior is observed when analyzing the performance over the Twitter dataset which contains shorter documents. This suggests that the proposed method has a high and stable performance under short and long temporal dependencies in texts for sentiment classification. This may be due to the fact that the resulting vector formed by 0, 1, and -1 better simplifies the contribution of each word's polarity to capture the overall polarity in texts, contrasting with more complex word representations originally proposed for content classification. Additionally, the nonlinear mapping of the polarity of each word to a continuous value by using all of its surrounding scores eases the final discrimination task of the network. This claim is supported by the fact that no improvement is achieved when a recurrent architecture is added to a feed-forward network, but when we use a bidirectional recurrent model instead, that maps each binary polarity considering all the surrounding polarities within the document, a noticeable increase in performance is consistently observed.

In summary, Senti-Sequence performs well over short and long text collections, without needing large amounts of text data. The experimental results suggest that the nonlinear mapping strategy enabled by a bidirectional recurrent model over word polarities is able to capture the overall sentiment of a text document expressed as a binary value. An important part of this mechanism relies on the accurate polarity prediction made by a Support Vector Machine model trained over rich word embedding collections such as ConceptNet.

7. Conclusions and Future Work

Sentiment analysis in user-written documents is a challenging task. This task is paramount to understanding unstructured text content generated by users in different domains, such as shopping, entertainment platforms, and social networks. Most current approaches deal with classic representations of the data, thus not capturing the complexity of the statements made in the comments.

In this work, we propose a novel representation that performs well with different models and lengths of documents. The approach consists of creating a new sentiment-aware document representation based on the output of a binary classifier that labels the polarity of each word in the document. Our approach attains a very high and consistent performance in the experiments, which is also observed when comparing it against state-of-the-art solutions over one of the employed datasets. We identified that a potential shortcoming of this approach could be its performance over text documents having labels in categories with more than two values.

We plan to continue working on the algorithm for the multiclass scenario. In this setting, we want to study the integration of attention mechanisms to limit the effect of surrounding words in the nonlinear polarity mapping in addition to its utility to weigh the effect of each word's polarity to the overall sentiment score. Finally, we also believe that adding explainability to the predictions must also be an important future goal to this work.

Author Contributions: A.R.M. performed all the experimentation and data processing tasks. H.A.-C. and J.Z. contributed to writing and the methodology design. The three authors worked together on the design of the proposed algorithm. All authors have read and agreed to the published version of the manuscript.

Funding: The work of Héctor Allende-Cid was funded by project 039.457/2020 VRIEA of Pontificia Universidad Católica de Valparaíso, Chile. The work of Juan Zamora was partially funded by project Fondecyt Initiation into Research 11200826 by ANID, Chile. The APC was funded by Lamarr Institute for Machine Learning and Artificial Intelligence.

Data Availability Statement: The IMDB reviews dataset was introduced in [49] and can be found at https://ai.stanford.edu/~amaas/data/sentiment/. The Twitter US Airline Sentiment dataset was originally collected by Crowdflower's Data for Everyone library and made available on Kaggle as a dataset at https://www.kaggle.com/datasets/crowdflower/twitter-airline-sentiment/data. The Amazon Books review is a subset of the Amazon Review Data (2018) dataset, introduced in [50] and can be found at https://cseweb.ucsd.edu/~jmcauley/datasets/amazon_v2/. SentiwordNet lexicon can be found at https://github.com/aesuli/SentiWordNet. SenticNet lexicon can be found at https://sentic.net. SentiWords lexicon can be found at https://nlplab.fbk.eu/tools-and-resources/lexical-resources-and-corpora/sentiwords. NLTK sentiment package was also used and it can be found at https://www.nltk.org/api/nltk.sentiment.html. Positive and Negative words can be found at https://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html (all accessed on 15 January 2024).

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

References

1. Nasukawa, T.; Yi, J. Sentiment analysis: Capturing favorability using natural language processing. In Proceedings of the 2nd International Conference on Knowledge Capture, Sanibel Island, FL, USA, 23–25 October 2003; pp. 70–77. [CrossRef]
2. Kumar, A.; Sebastian, T.M. Sentiment analysis: A perspective on its past, present and future. *Int. J. Intell. Syst. Appl.* **2012**, *4*, 1–14. [CrossRef]
3. Šaloun, P.; Hruzik, M.; Zelinka, I. Sentiment analysis-e-bussines and e-learning common issue. In Proceedings of the 2013 IEEE 11th International Conference on Emerging eLearning Technologies and Applications (ICETA), Stary Smokovec, Slovakia, 24–25 October 2013; pp. 339–343.
4. Ishijima, H.; Kazumi, T.; Maeda, A. Sentiment analysis for the Japanese stock market. *Glob. Bus. Econ. Rev.* **2015**, *17*, 237–255. [CrossRef]
5. Denecke, K.; Deng, Y. Sentiment analysis in medical settings: New opportunities and challenges. *Artif. Intell. Med.* **2015**, *64*, 17–27. [CrossRef]
6. Sarlan, A.; Nadam, C.; Basri, S. Twitter sentiment analysis. In Proceedings of the 6th International Conference on Information Technology and Multimedia, Putrajaya, Malaysia, 18–20 November 2014; pp. 212–216.
7. Kang, H.; Yoo, S.J.; Han, D. Senti-lexicon and improved Naïve Bayes algorithms for sentiment analysis of restaurant reviews. *Expert Syst. Appl.* **2012**, *39*, 6000–6010. [CrossRef]
8. Moraes, R.; Valiati, J.F.; Neto, W.P.G. Document-level sentiment classification: An empirical comparison between SVM and ANN. *Expert Syst. Appl.* **2013**, *40*, 621–633. [CrossRef]
9. Bhatia, P.; Ji, Y.; Eisenstein, J. Better Document-level Sentiment Analysis from RST Discourse Parsing. *arXiv* **2015**, arXiv:1509.01599.
10. Wu, C.; Wu, F.; Liu, J.; Huang, Y.; Xie, X. Sentiment lexicon enhanced neural sentiment classification. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management, Beijing, China, 3–7 November 2019; pp. 1091–1100.

11. Thongtan, T.; Phienthrakul, T. Sentiment classification using document embeddings trained with cosine similarity. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*; Association for Computational Linguistics: Florence, Italy, 2019; pp. 407–414.
12. Yousif, A.; Niu, Z.; Tarus, J.K.; Ahmad, A. A survey on sentiment analysis of scientific citations. *Artif. Intell. Rev.* **2019**, *52*, 1805–1838. [CrossRef]
13. Anees, A.F.; Shaikh, A.; Shaikh, A.; Shaikh, S. Survey Paper on Sentiment Analysis: Techniques and Challenges. Technical Report, EasyChair. 2020. Available online: https://easychair.org/publications/preprint/Sc2h (accessed on 15 January 2024).
14. Yue, L.; Chen, W.; Li, X.; Zuo, W.; Yin, M. A survey of sentiment analysis in social media. *Knowl. Inf. Syst.* **2019**, *60*, 617–663. [CrossRef]
15. Zhang, L.; Wang, S.; Liu, B. Deep learning for sentiment analysis: A survey. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2018**, *8*, e1253. [CrossRef]
16. Pang, B.; Lee, L.; Vaithyanathan, S. Thumbs up? Sentiment classification using machine learning techniques. *arXiv* **2002**, arXiv:cs/0205070.
17. Pak, A.; Paroubek, P. Twitter as a corpus for sentiment analysis and opinion mining. In Proceedings of the International Conference on Language Resources and Evaluation, LREC 2010, Valletta, Malta, 17–23 May 2010; Volume 10, pp. 1320–1326.
18. 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, Massachusetts Institute of Technology, Cambridge, MA, USA, 9–11 October 2010; pp. 1046–1056.
19. 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, Lake Tahoe, NV, USA, 5–10 December 2013; pp. 3111–3119.
20. Tang, D.; Wei, F.; Qin, B.; Yang, N.; Liu, T.; Zhou, M. Sentiment embeddings with applications to sentiment analysis. *IEEE Trans. Knowl. Data Eng.* **2015**, *28*, 496–509. [CrossRef]
21. Ren, Y.; Zhang, Y.; Zhang, M.; Ji, D. Improving twitter sentiment classification using topic-enriched multi-prototype word embeddings. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016; p. 7.
22. Yu, L.C.; Wang, J.; Lai, K.R.; Zhang, X. Refining word embeddings for sentiment analysis. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, 7–11 September 2017; pp. 534–539.
23. Xiong, S.; Lv, H.; Zhao, W.; Ji, D. Towards Twitter sentiment classification by multi-level sentiment-enriched word embeddings. *Neurocomputing* **2018**, *275*, 2459–2466. [CrossRef]
24. Johnson, R.; Zhang, T. Effective Use of Word Order for Text Categorization with Convolutional Neural Networks. In Proceedings of the 2015 Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, CO, USA, 31 May–5 June 2015; pp. 103–112.
25. Taboada, M.; Brooke, J.; Tofiloski, M.; Voll, K.; Stede, M. Lexicon-based methods for sentiment analysis. *Comput. Linguist.* **2011**, *37*, 267–307. [CrossRef]
26. Muhammad, A.; Wiratunga, N.; Lothian, R. Contextual sentiment analysis for social media genres. *Knowl.-Based Syst.* **2016**, *108*, 92–101. [CrossRef]
27. Asghar, M.Z.; Khan, A.; Ahmad, S.; Qasim, M.; Khan, I.A. Lexicon-enhanced sentiment analysis framework using rule-based classification scheme. *PLoS ONE* **2017**, *12*, e0171649. [CrossRef]
28. Wang, J.H.; Liu, T.W.; Luo, X.; Wang, L. An LSTM approach to short text sentiment classification with word embeddings. In Proceedings of the 30th Conference on Computational Linguistics and Speech Processing (ROCLING 2018), Hsinchu, Taiwan, 4–5 October 2018; pp. 214–223.
29. Strapparava, C.; Valitutti, A. Wordnet affect: An affective extension of wordnet. In Proceedings of the Fourth International Conference on Language Resources and Evaluation, LREC 2004, Lisbon, Portugal, 26–28 May 2004; Volume 4, p. 40.
30. Esuli, A.; Sebastiani, F. Sentiwordnet: A publicly available lexical resource for opinion mining. In Proceedings of the Fifth International Conference on Language Resources and Evaluation, LREC 2006, Genoa, Italy, 22–28 May 2006; Volume 6, pp. 417–422.
31. Shin, B.; Lee, T.; Choi, J.D. Lexicon Integrated CNN Models with Attention for Sentiment Analysis. In Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, Copenhagen, Denmark, 8 September 2017; pp. 149–158.
32. Lei, Z.; Yang, Y.; Yang, M. Sentiment lexicon enhanced attention-based LSTM for sentiment classification. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; pp. 1–2.
33. Li, W.; Qi, F.; Tang, M.; Yu, Z. Bidirectional LSTM with self-attention mechanism and multi-channel features for sentiment classification. *Neurocomputing* **2020**, *387*, 63–77. [CrossRef]
34. Li, W.; Zhu, L.; Shi, Y.; Guo, K.; Zheng, Y. User reviews: Sentiment analysis using lexicon integrated two-channel CNN-LSTM family models. *Appl. Soft Comput.* **2020**, *94*, 106435. [CrossRef]
35. Wang, X.; Liu, Y.; Sun, C.J.; Wang, B.; Wang, X. Predicting polarities of tweets by composing word embeddings with long short-term memory. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, ACL 2015, Beijing, China, 26–31 July 2015; Volume 1: Long Papers , pp. 1343–1353.

36. Mao, X.; Chang, S.; Shi, J.; Li, F.; Shi, R. Sentiment-aware word embedding for emotion classification. *Appl. Sci.* **2019**, *9*, 1334. [CrossRef]
37. Baccianella, S.; Esuli, A.; Sebastiani, F. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In Proceedings of the International Conference on Language Resources and Evaluation, LREC 2010, Valletta, Malta, 17–23 May 2010; Volume 10, pp. 2200–2204.
38. Cambria, E.; Poria, S.; Hazarika, D.; Kwok, K. SenticNet 5: Discovering conceptual primitives for sentiment analysis by means of context embeddings. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; p. 8.
39. Gatti, L.; Guerini, M.; Turchi, M. SentiWords: Deriving a high precision and high coverage lexicon for sentiment analysis. *IEEE Trans. Affect. Comput.* **2015**, *7*, 409–421. [CrossRef]
40. Loper, E.; Bird, S. NLTK: The natural language toolkit. *arXiv* **2002**, arXiv:cs/0205028.
41. Hu, M.; Liu, B. Mining and summarizing customer reviews. In Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, WA, USA, 22–25 August 2004; pp. 168–177.
42. Liu, G.; Guo, J. Bidirectional LSTM with attention mechanism and convolutional layer for text classification. *Neurocomputing* **2019**, *337*, 325–338. [CrossRef]
43. Speer, R.; Chin, J.; Havasi, C. ConceptNet 5.5: An Open Multilingual Graph of General Knowledge. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; AAAI Press: Palo Alto, CA, USA, 2017; Volume 31.
44. Moen, S.; Ananiadou, T.S.S. Distributional semantics resources for biomedical text processing. In Proceedings of the 5th International Symposium on Languages in Biology and Medicine (LBM 2013), Tokyo, Japan, 12–13 December 2013; pp. 39–44.
45. Turc, I.; Chang, M.W.; Lee, K.; Toutanova, K. Well-read students learn better: On the importance of pre-training compact models. *arXiv* **2019**, arXiv:1908.08962.
46. Haonan, L.; Huang, S.H.; Ye, T.; Xiuyan, G. Graph star net for generalized multi-task learning. *arXiv* **2019**, arXiv:1906.12330.
47. Sachan, D.S.; Zaheer, M.; Salakhutdinov, R. Revisiting lstm networks for semi-supervised text classification via mixed objective function. In Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 6940–6948.
48. Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J.; Salakhutdinov, R.; Le, Q.V. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv* **2019**, arXiv:1906.08237.
49. Maas, A.; Daly, R.; Pham, P.; Huang, D.; Ng, A.; Potts, C. Learning Word Vectors for Sentiment Analysis. In Proceedings of the 49th Annual Meeting of the Association For Computational Linguistics: Human Language Technologies, Portland, OR, USA, 19–24 June 2011; pp. 142–150.
50. Ni, J.; Li, J.; McAuley, J. Justifying Recommendations using Distantly-Labeled Reviews and Fine-Grained Aspects. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, 3–7 November 2019; pp. 188–197.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.