applied sciences

MDPI

Article

Examining Attention Mechanisms in Deep Learning Models for Sentiment Analysis

Spyridon Kardakis [1], Isidoros Perikos [1,2,*], Foteini Grivokostopoulou [1,2] and Ioannis Hatzilygeroudis [1]

[1] Computer Engineering and Informatics Department, University of Patras, 26504 Patras, Greece; kardakis@ceid.upatras.gr (S.K.); grivokwst@ceid.upatras.gr (F.G.); ihatz@ceid.upatras.gr (I.H.)
[2] Computer Technology Institute and Press "Diophantus", 26504 Patras, Greece
* Correspondence: perikos@ceid.upatras.gr

Abstract: Attention-based methods for deep neural networks constitute a technique that has attracted increased interest in recent years. Attention mechanisms can focus on important parts of a sequence and, as a result, enhance the performance of neural networks in a variety of tasks, including sentiment analysis, emotion recognition, machine translation and speech recognition. In this work, we study attention-based models built on recurrent neural networks (RNNs) and examine their performance in various contexts of sentiment analysis. Self-attention, global-attention and hierarchical-attention methods are examined under various deep neural models, training methods and hyperparameters. Even though attention mechanisms are a powerful recent concept in the field of deep learning, their exact effectiveness in sentiment analysis is yet to be thoroughly assessed. A comparative analysis is performed in a text sentiment classification task where baseline models are compared with and without the use of attention for every experiment. The experimental study additionally examines the proposed models' ability in recognizing opinions and emotions in movie reviews. The results indicate that attention-based models lead to great improvements in the performance of deep neural models showcasing up to a 3.5% improvement in their accuracy.

Keywords: attention mechanism; deep neural networks; global-attention; hierarchical-attention; self-attention; sentiment analysis

Citation: Kardakis, S.; Perikos, I.; Grivokostopoulou, F.; Hatzilygeroudis, I. Examining Attention Mechanisms in Deep Learning Models for Sentiment Analysis. *Appl. Sci.* **2021**, *11*, 3883. https://doi.org/10.3390/app11093883

Received: 2 April 2021
Accepted: 23 April 2021
Published: 25 April 2021

1. Introduction

In the past decade, the dramatic decrease in computation cost and the drastic increase in data availability led to the emergence of a new sub-field of machine learning, called deep learning, which outperforms its predecessor by achieving very high performance on large data. Deep learning, utilizes deep artificial neural networks, such as CNNs, RNNs [1], LSTMs [2] and GRUs [3], which achieve remarkable performance in various domains, such as speech recognition [4–6], signal and EEG analysis [7], computer vision [8–10], emotion recognition [11–13], disease and cancer recognition [14] as well as in text classification and sentiment analysis [15–17]. Indeed, in-text classification and sentiment analysis, deep neural networks have demonstrated quite remarkable performance [18]. Despite their interpretability disadvantages and their computational cost, deep neural networks can model complex nonlinear relationships. The extra layers enable the composition of features from lower layers, potentially modeling complex data with fewer units than a similarly performing shallow network [19].

Attention is arguably one of the most powerful recent concepts in deep learning, bringing out many breakthroughs in the field and improving the performance of neural networks [20]. An attention mechanism effectively gives a model the ability to "attend to" a certain part of the input sequence, which would arguably be a part with higher importance [21]. For example, it allows an LSTM to focus on specific parts of a document or sentences [22], as shown in Figure 1.

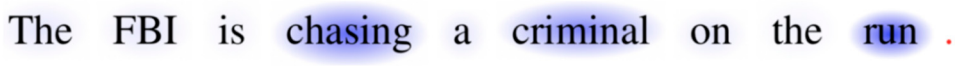


Figure 1. Illustration of word weighting of past memory using an attention mechanism [22].

Attention-based models have been used in various topics, including machine translation, speech recognition, and sentiment classification [23,24]. The field of Sentiment analysis, in particular, has been met with increased interest by the research community in recent years due to the rise of social media and the Internet. The vast amount of opinion-heavy user-generated content available to us, whether that is a product review or an opinion on an event, shows that effective sentiment analysis is needed. In particular, sentiment analysis is the computational study of people's opinions, emotions and attitudes towards entities, such as products, services and events [25]; it is highly sought after by businesses and service providers. However, automatic knowledge extraction about people's opinions and emotional states can be a very challenging task [26]. As a result, achieving performance gains, even on a small scale is of vital importance in natural language processing and sentiment analysis [27].

Overall, in this work we present various attention-based models and study their performance under different training methods and topologies for sentiment analysis. Specifically, we present attention-based models based on recurrent neural networks (RNNs) and examine their functionality in various contexts and data. A comparative analysis is performed in a text classification task where for every experiment, baseline neural models are compared with and without the use of an attention mechanism. The experimental study additionally examines the performance of the proposed methods in recognizing opinions and emotions in textual data, utilizing widely used benchmark datasets. The results indicate that attention-based models lead to si gains in performance of up to 3.5% in terms of accuracy. It is worth noting that attention-based models can weigh the importance of certain words, an ability that is quite limited in conventional neural network models. The main focus and contribution of this work is to present and implement self-attention, global-attention and hierarchical-attention-based deep neural networks, that can recognize sentiment in text and then thoroughly examine their performance in various contexts and under different baseline models, training hyperparameters and architectures.

The rest of this paper is structured as follows: Section 2 reviews the related work on attention-based methods and sentiment analysis. Sections 3 and 4 introduce the theory and the mathematical background behind the aforementioned field. The experimental study and the results collected are presented in Section 5. After this, Section 6 discusses the results and presents the main findings of the study. Finally, Section 7 provides conclusions as well as directions for future work on attention-based techniques.

2. Related Works

Attention-based methods have found many applications in various natural language processing tasks, including sentiment analysis, opinion mining as well as in recognition of emotions in text. Bahdanau et al. [23] present an encoder–decoder approach with an attention mechanism that is integrated into the decoder by repeatedly reading the representation of a source sentence, which remains fixed after being generated by the encoder. Thus, the model can search for parts that are relevant to predicting a target word and, as a result, achieved state-of-the-art performance.

Munkhdalai et al. [28] propose a middle ground between two popular deep neural network architectures, recurrent and recursive neural networks. The approach named neural tree indexer (NTI) is a syntactic parsing-independent tree-structured model that learns a premise and a hypothesis representation and then combines them with an attention mechanism, achieving a new state-of-the-art accuracy on inference and classification tasks. Yang et al. [29] propose a model named hierarchical attention network (HAN) for a document classification task that mirrors the hierarchical structure of documents. Furthermore, since words and sentences vary in terms of informativeness, it has two levels of attention

mechanisms applied at the word and sentence level. As a result, it can differentiate its attention to more and less important content when constructing the document representation while also using context to discover when a sequence is relevant. The encoder used is a GRU, and the proposed approach outperforms previous methods on six datasets. Yin et al. [30] expand upon the previous model and propose a hierarchical iterative attention model for an aspect-based sentiment analysis task. The aspect-based task is formulated as a machine comprehension problem. The model achieves greater performance than the hierarchical attention network baseline. Lee et al. [31] propose a method for identifying keywords discriminating positive and negative sentences by using a weakly supervised learning method based on a CNN. The CNN model is trained on sentence matrices, and after the training, a word attention mechanism is employed that identifies high-contributing words with a class activation map (CAM), using the weights from the fully connected layer at the end of the CNN. The main advantage is that it produces both sentence-level polarity scores and word-level polarity scores by only using weak labels, i.e., the polarity of the sentence from the dataset.

Lin et al. [32] propose a model for extracting sentence embedding by introducing self-attention. In this new sentence embedding technique, each embedding corresponds to a 2-D matrix, with each row of the matrix attending on a different part of the sentence. First, the sentence is run through an RNN. Then multiple attention values are learned for each RNN state, and then each attention vector focuses on different parts of the sentence because a penalization term is added. It achieves high accuracy on both sentiment classification and textual entailment tasks. Chen et al. [33] propose an LSTM model, which, unlike existing methods, incorporates global user preference and product characteristics on a sentiment classification task. The model is a hierarchical LSTM, and it generates sentence and document representations. Then, an attention mechanism is used on the user and product information, thus taking into consideration information at both the word level and the semantic level.

Wang et al. [24] propose an attention-based LSTM method with target embedding for an aspect-level sentiment classification task. The attention mechanism forces the neural model to attend to the important part of a sentence when different aspects are taken as input. The authors show two ways to consider aspect information during attention: (1) to concatenate the aspect vector into the sentence hidden representations for computing attention weights (2) to additionally append the aspect vector into the input word vectors. ATAE-LSTM achieves state-of-the-art performance on a SemEval dataset. Liu et al. [34] propose a BiLSTM with an attention mechanism to focus on the information output from the hidden layers, combined with a convolutional layer that extracts higher-level phrase representations. Fu et al. [35] extend the traditional LSTM-based approach for sentiment analysis by using a lexicon. Overall, the lexicon enhances the ability of word embeddings to represent words, while an attention mechanism is also introduced, which utilizes the global information of the entire text instead of having a specific target.

Dou et al. [36] propose a deep memory network for document-level sentiment classification. The proposed model can capture both the user and product information at the same time. A memory network consists of several inference components combined with a large long-term memory while also utilizing the memory as a knowledge base. The architecture of the model consists of two parts. First, an LSTM is used to represent each document, then a deep memory network consisting of multiple layers (hops) is used to predict the ratings for each document, and each layer is a content-based attention model.

Li et al. [37] propose an adversarial memory network (AMN) for cross-domain sentiment analysis. In domain adaptation tasks, the term "pivot" is used, which operates similarly to discriminative learning. The end-to-end AMN can automatically capture the pivots using an attention mechanism. The proposed approach is evaluated against various techniques, achieving a 4.36% better accuracy. Tang et al. [38] propose an end-to-end memory network for an aspect-level sentiment classification task. It can capture the importance of each context word when inferring the sentiment polarity of an aspect. An attention

mechanism is employed with external memory to capture the importance of each context word concerning the given target aspect. Multiple computational layers are applied to the input text representation, each of which is a neural attention model with external memory, while the entire model can be trained end-to-end with gradient descent.

Shuang et al. [39] found that the keywords which express the sentiment for the aspect word to a maximum degree are always close to the aspect word itself. To this end, two parameters are designed for both sides according to their respective importance, which is also to be learned through training. Directly inputting the word vector into the model and generating the representations of the sentence vector will contain information that is not related to the aspect word. Therefore, they use the location weight to emphasize the related sentiment information towards aspect words in the process of generating sentence vectors. Given the fact that location information is beneficial to obtain a better result, input word embedding vectors are weighted by location weights. Since they expect the weights on words far away from the aspect word to fall faster, which helps to prevent the interference of information that is not related to aspects, they use the Laplacian probability density function to obtain the location weights. The authors report quite interesting results and indicate an accuracy of 0.8035 in the SemEval 2015 task 4 dataset.

Liu et al. [34] propose a deep neural network architecture for a text classification task. The proposed model is named attention-based bidirectional long short-term memory with convolution layer (AC-BiLSTM). In particular, the first layer operates as a phrase representation based on word embeddings, followed by a BiLSTM that accesses context representations alongside an attention mechanism. AC-BiLSTM is evaluated on sentiment analysis datasets against state-of-the-art methods, outperforming them. The authors conclude that for the convolutional layer, the convolution window size and the stride size affect the classification performance and that the BiLSTM showcases better performance than the convolution layer.

Chen et al. [40] propose a deep learning system specifically for short texts since such a task is faced with particular challenges, such as lack of contextual information. The main idea is to utilize external knowledge to enhance the representations by making use of embeddings. Conceptual information is used to acquire the weight of concepts from two aspects, with attention acting like a human being who has an intrinsic ability to pay attention to important knowledge. The proposed system is named deep short text classification with knowledge-powered attention (STCKA). It is evaluated against other models and achieved quite interesting performance.

3. Deep Recurrent Neural Networks

As a base for attention-based deep neural networks, we utilize recurrent neural networks (RNNs). A recurrent neural network is an extension of the conventional feed-forward neural network. Long short-term memory (LSTM) models, which were introduced by Hochreiter et al. [2], are also based on the RNN architecture. They overcome RNN's shortcomings when it comes to gradients and also improve the learning ability for long-time sequence data. The differences are that instead of having a single neural network layer, there are four layers interacting in a specific way. The LSTM unit is composed of a cell, an input gate, an output gate, and a forget gate. The memory cell that is introduced can preserve state over long periods of time, while the three gates regulate the flow of information into and out of the cell:

$$X = [h_{t-1}, x_t] \tag{1}$$

$$f_t = \sigma\Big(W_f \cdot X + b_f\Big) \tag{2}$$

$$i_t = \sigma(W_i \cdot X + b_i) \tag{3}$$

$$o_t = \sigma(W_o \cdot X + b_o) \tag{4}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot tanh(W_c \cdot X + b_c) \tag{5}$$

$$h_t = o_t \odot tanh(c_t) \tag{6}$$

where $W_i, W_f, W_o \in \mathbb{R}^{d \times 2d}$ are the weighted matrices, $b_i, b_f, b_o \in \mathbb{R}^{d}$ are the biases of LSTM to be learned during training, σ is the sigmoid function, x_t includes the inputs representing the word embedding vectors, and h_t is the vector of the hidden layer. Overall o_t decides the output, using a sigmoid function on the "output gate".

Gated recurrent units (GRUs) are a variation of LSTMs introduced by Cho et al. [3]. They include a gating mechanism and combine the "forget" and "input" gates into a signal update gate as well as performing some other changes, resulting in a model simpler than LSTMs:

$$z_t = \sigma(W_z \cdot X + b_z) \tag{7}$$

$$r_t = \sigma(W_r \cdot X + b_r) \tag{8}$$

$$\hat{h}_t = tanh(W_h \cdot (r_t \odot X)) \tag{9}$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \hat{h}_t \tag{10}$$

where z_t is the update gate vector, r_t is the reset gate vector and h_t is the candidate activation vector.

The main difference between a recurrent neural network and a deep recurrent neural network is that in the latter, we stack multiple layers of individual recurrent networks. The idea for this implementation came from the problem that, while RNNs have depth, this notion of depth is unlikely to involve a hierarchical interpretation of the data. By applying the same computation recursively to compute the contribution of children to their parents and the same computation to produce an output response, every node/phrase is being represented in the same space. This technique is different than conventionally stacking deep learners, where an important benefit of depth is the hierarchy among hidden representations: every hidden layer conceptually lies in a different representation space and potentially is a more abstract representation of the input than the previous layer [19].

In the context of this work, we utilize two different representations for the input text, which are the one-hot encoding and the word2vec embeddings, and also examine their impact on attention-based methods. Word embeddings try to capture the characteristics of the neighbors of the target and, as a result, capture the similarity between words. These vector representations of words map a word into a common space preserving the word meaning, their relationships and the semantic information. The word2vec method [41] that was utilized in this work implements both the continuous bag-of-words (CBOW) and the skip-gram models to learn word embeddings. As mentioned above, word2vec is a fixed dimensional real-valued vector representation that transforms an atomic word into a positional representation of the word relative to other words in the dataset. We also examine the impact of the dropout regularization technique [42]. Dropout aims to reduce over-fitting by preventing complex adaptations during training and excluding rare dependencies from the model. In addition, it creates subnetworks (child models), aiming to minimize the expected loss by dropping out units.

Furthermore, regarding the optimization strategy used to update the parameters of our model during training, we utilized the Adam optimizer [43], known for its robust performance across various tasks. Adam computes and adapts learning rates specifically to individual features. Contrary to other methods, it calculates current gradients by storing a decaying average of past gradients while also utilizing momentum. The Adam algorithm features fast convergence, low variance and a healthy learning rate that does not vanish. In the experimental study, the parameters of the training procedure are presented in detail.

4. Attention-Based Methods

Although RNNs have been a great success, they have certain limitations [44]. Among them, they are not efficient at memorizing long or distant sequences [45]. RNNs process tokens sequentially, maintaining a state vector representing the data seen after every token. The information from a token can propagate arbitrarily down the sequence by continuously

encoding information. However, the main issue that arises here is that due in part to the vanishing gradient problem [44], the model's state at the end of a long sentence often does not contain information about early tokens and as a result, the mechanism does not work as intended.

The aforementioned problem was addressed by the introduction of attention mechanisms. Attention mechanisms let a model directly look at and draw from the state of an earlier point in the sentence. The attention layer can access all previous states and weigh them according to some learned measure of relevancy to the current token, providing sharper information about far-away relevant tokens. A clear example of the utility of attention is in neural machine translation [23]. In an English-to-French translation system, the first word of the French output most probably depends heavily on the beginning of the English input. Though, in a classic LSTM model, to produce the first word of the French output, the model is only given the state vector of the last English word. In theory, this vector can encode information about the whole English sentence, but as mentioned before, this information is often not preserved in practice. If an attention mechanism is introduced, the model can instead learn to attend to the states of early English tokens when producing the beginning of the French output, giving it a much better concept of what it is translating.

Let $H \in \mathbb{R}^{d \times N}$ be a matrix consisting of hidden vectors $[h_1, \ldots, h_N]$ that the LSTM produced, where d is the size of hidden layers. Furthermore, v_α represents the embedding. The attention mechanism will produce an attention weight vector α and a weighted hidden representation r:

$$M = tanh\left(\begin{bmatrix} W_h H \\ W_v v_\alpha \otimes e_N \end{bmatrix}\right) \tag{11}$$

$$a = softmax\left(W^T M\right) \tag{12}$$

$$r = Ha^T \tag{13}$$

where W are the weights to be learned during training. The attention mechanism allows the model to capture the most important parts of a document. Using a hyperbolic tangent on the representation would lead to a more complex feature representation $h^* \in \mathbb{R}^d$ that would be more fitting for other tasks. Finally, we can get the sentiment distribution of each text by adding a softmax layer in each final text representation r:

$$y = softmax(w \cdot r + b) \tag{14}$$

This type of attention is also known as global (soft) attention [23]. As we can see in Figure 2, at each time step t, a global attentional model infers a variable-length alignment weight vector α_t based on the current target state h_t and all source states $\bar{h}_s$ [46]. A global context vector c_t is then computed as the weighted average, according to α_t, over all the source state. The idea is to derive a context vector based on all hidden states of the encoder RNN. Hence, it is said that this type of attention attends to the entire input state space.

An alternative attention mechanism is self-attention [22]. The main goal is to overcome the drawback of RNNs by allowing the attention mechanism to focus on segments of the sentence, where the relevance of the segment is determined by the contribution to the task. Different positions of the same hidden state space derived from the input sequence are related to different positions, based on the argument that multiple components together form the overall semantics of a sequence. This approach brings together differently positioned information through multiple hops' attention. In Figure 3, the architecture of the self-attention model is illustrated [32]. The embedded tokens are fed into LSTM layers (h_i). Hidden states are weighted by an attention vector (A_{ij}) to obtain a refined sentence representation that is used as an input for the classification. To acquire the weights, we multiply the hidden states by a weight matrix and then perform a series of transformations. With $2u$ being the dimension of the hidden state of a layer, a linear transformation is applied to the n hidden state vectors from $2u$-dimensional space to a d-dimensional one. After applying hyperbolic tangent activation, another linear transformation from d-dimension

to *r*-dimension is applied to come up with *r* dimensional attention vector per token. As a result, we have *r* attention weight vectors of size *n*.

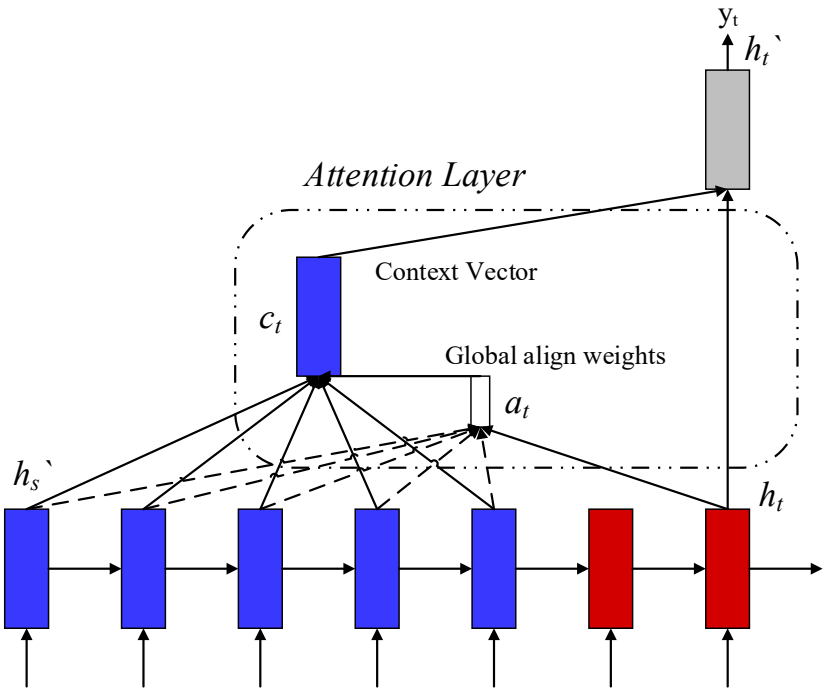


Figure 2. The structure and functionality of a global-attention mechanism.

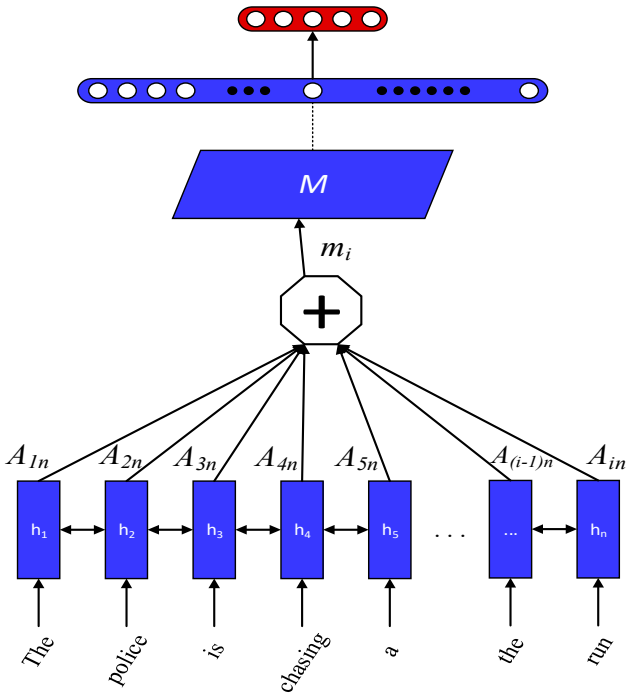


Figure 3. The structure and functionality of a self-attention mechanism.

A penalty term is used based on the self-attention matrix to prevent multiple attention vectors from being similar or redundant. This is done by regularizing the similarity of r-hops so that the attention mechanism does not always provide similar annotation weights:

$$P = \|(AA^T - I)\|_F^2 \quad (15)$$

Another attention mechanism is hierarchical attention [29]. The main idea is to reflect the hierarchical structure that exists within documents as illustrated in Figure 4 [29]. A bottom-up approach is followed by applying attention mechanisms sequentially at word and sentence levels, but a top-down approach (ex. word and character levels) is also applicable. Thus, this type of mechanism is said to attend differentially to more and less important content when constructing the document representation.

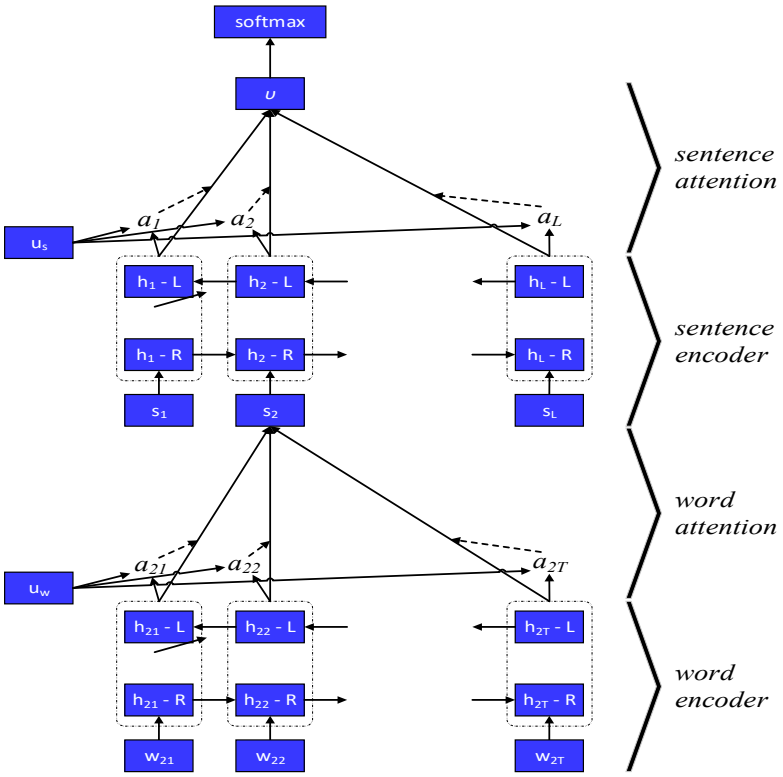


Figure 4. The structure and functionality of a hierarchical-attention network (HAN).

As the base of the entire approach, a traditional neural network is used on two levels, first as the word encoder and second as the sentence encoder; in most cases, that is a bidirectional recurrent neural network, for example, a Bi-LSTM. Then, a word attention mechanism is introduced, which can extract words that are important to the meaning of the sentence and aggregate the representation of those informative words to form a sentence vector. Those "improved" annotations are then represented by u_{it}:

$$u_{it} = tanh(W_w h_{it} + b_w) \tag{16}$$

Then, the annotations are again multiplied with a trainable context vector u_w and normalized to an importance weight per word a_{it} by a softmax function as shown in Figure 4:

$$a_{it} = \frac{exp(u_{it}^T u_w)}{\sum_t exp(u_{it}^T u_w)} \tag{17}$$

$$s_i = \sum_t a_{it} h_{it} \tag{18}$$

Finally, the sum of these importance weights concatenated with the previously calculated context annotations forms a sentence vector. Furthermore, a sentence attention mechanism follows the word attention one, operating in the same way. Trainable weights

are again jointly learned during the training process, and the final output is a document vector u, which can be used as a feature vector for the document classification:

$$u = \sum_i a_i h_i \tag{19}$$

5. Experimental Study

A concrete experiment was designed to study the functionality of various attention methods and thoroughly examine their performance. Publicly available benchmark datasets were utilized, featuring different types of textual data, aiming to assess the performance of self-attention, global-attention and hierarchical-attention methods in order to provide a deeper insight into their performance on heterogeneous data.

5.1. Datasets

Reviews constitute the basis for almost all benchmark sentiment analysis datasets in the literature used to evaluate a model's performance [47]. The proposed attention-based methods are evaluated on three sentiment analysis datasets, the movie review polarity (MR) [48], the movie review subjectivity (SUBJ) [49] and the IMDb large movie review dataset [50].

The movie review polarity dataset (MR) is a collection of movie reviews extracted from the website Rotten Tomatoes in 2004. This dataset comes in two different versions, polarity v2.0, which consists of 1000 positive and 1000 negative processed reviews and sentence polarity v1.0, which consists of 5331 positive and 5331 negative processed sentences/snippets. All the reviews are written before 2002, with a cap of 20 reviews per author (312 authors in total) per category. What makes the dataset particularly interesting is the informal language used since each instance is very short and the presence of noisy polarity labeling.

The movie review subjectivity dataset (SUBJ) was constructed in 2004 by crawling two popular movie review websites to tackle the problem of building a subjectivity corpus. To gather subjective sentences (or phrases), the authors collected 5000 movie review snippets (short text) from the website Rotten Tomatoes. To obtain (mostly) objective data, the authors took 5000 sentences from plot summaries available from IMDb. It worth noting that only snippets of at least ten words and only reviews/summaries of movies released post-2001 were selected, which prevents overlap with the movie review polarity dataset.

The IMDb large movie review dataset (IMDB) is a collection of 50,000 labeled and 50,000 unlabeled reviews from the website IMDb and was constructed in 2011. It is much bigger than the previous datasets and does not have many correlations between the training the testing set. It was constructed without allowing more than 30 reviews per movie and contains an even number of positive and negative reviews. The authors decided to consider only highly polarized reviews, so reviews with a score equal or greater than 7 are labeled as positive, while reviews with a score equal or less than 4 are labeled as negative and neutral reviews are excluded. In this work, we used 25,000 documents for training and 25,000 for testing.

The purpose of the text classification task is to assign each document to a sentiment label, e.g., positive/negative/neutral. In the following section, the experimental setup, the parameters and the collected results are thoroughly presented.

5.2. Results

To evaluate the performance of the proposed models on a sentiment classification task, we compare every attention-based approach to equivalent baseline models that do not utilize attention mechanisms. Regarding the preprocessing of the data, the representation (embeddings) of the input documents is described in Section 3. The word2vec embedding vectors are pre-trained on an unlabeled corpus whose size is about 100 billion words. We trained all models with a batch size of 256 samples, a hidden neuron count of 256 and an initial learning rate of 0.001 for the Adam optimizer.

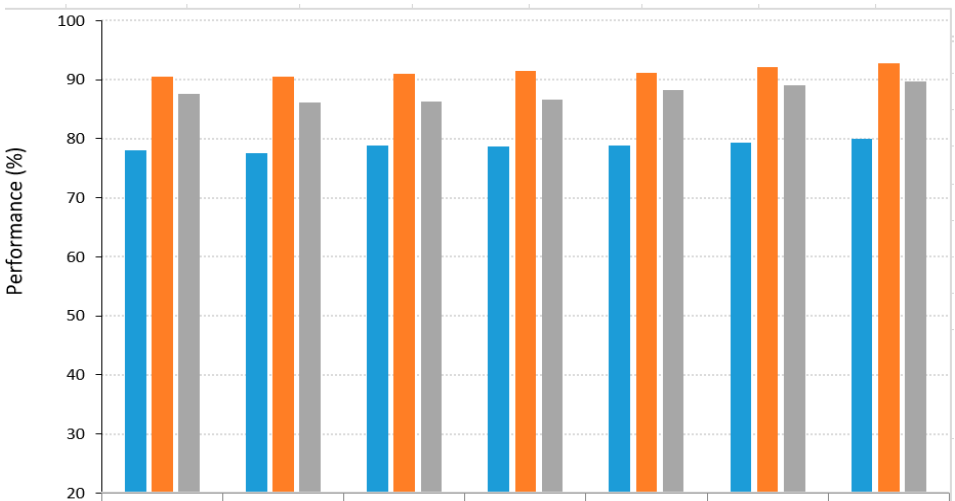
Overall, to implement the proposed models, we utilized the deep learning framework named Tensorflow [51]. We trained the models on an Nvidia (Santa Clara, CA, USA) RTX^{TM} 2080 Ti since Tensorflow supports GPU computations, which heavily speed up the training process.

To evaluate our models, we utilized k-fold cross-validation, which splits the data into k subsets of equal size. Then the entire process is executed k times in a row by selecting 1 subset as the test set and the remaining as the training set. This leads to k non-overlapping test data sets and k different accuracies for the model. Cross-validation ensures fairness in the testing process and reduces the probability of randomly selecting a biased/bad test set. We decided on a 10-fold cross-validation split, which is a rather popular configuration.

The models are evaluated on the commonly used performance metrics of accuracy and F1-score. They can be calculated as follows: In our experiment, if we define TP as the number of "true-positive" samples predicted as positive, FN as the number of true-positive samples predicted as negative, FP as the number of true-negative samples predicted as positive, and TN as the number of true-negative samples predicted as negative, then the metrics can be expressed as follows:

After various experiments, we come to certain conclusions regarding the best hyperparameters and settings. One of them is that using a pre-trained 300-dimensional word2vec and further refining it through additional training achieves the highest performance. Two experiments were designed and conducted, and each one examined the performance of different types of models.

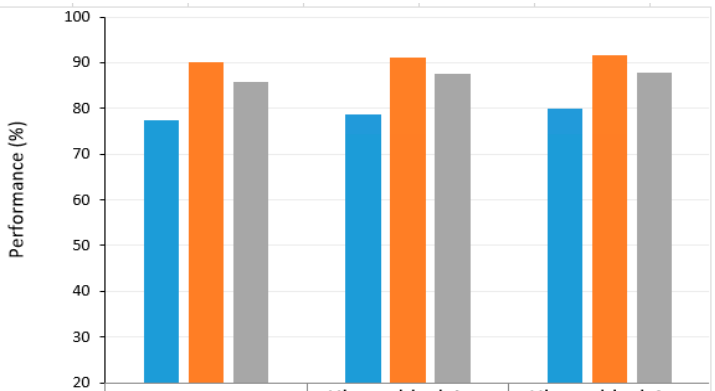
The first experiment assessed the performance of global and self-attention mechanisms using various models and parameters. Specifically, attention-based LSTMs with global attention (Global-Att-LSTM) and attention-based LSTMs with self-attention (Self-Att-LSTM) are implemented and assessed; their performance is compared to baseline models like CNNs, LSTMs as well as more sophisticated models like multiplicative and bidirectional LSTM models. In Figure 5, we illustrate the performance results of all the models examined in the context of the first experiment, thus illustrating a comparison between the different neural network models that were implemented and examined.



	CNN	LSTM	mLSTM	BiLSTM	Global-Att-LSTM	Self-Att-LSTM	Self-Att-LSTM+word2vec
MR	78.01	77.56	78.79	78.63	78.88	79.34	79.93
SUBJ	90.43	90.57	90.92	91.41	91.16	92.05	92.76
IMDB	87.54	86.21	86.25	86.69	88.29	89.12	89.71

Figure 5. Performance of the models on the SUBJ, MR and IMDB datasets.

The second experiment assessed the performance of hierarchical-attention mechanisms compared to baseline neural network models. Specifically, attention-based GRUs with hierarchical attention (Hierarchical-Att-GRU) are implemented and their performance is evaluated against baseline GRU models that do not feature an attention mechanism. In Figure 6, the performance results of all the aforementioned models are illustrated. Finally, an attention-based GRU that also incorporates dropout layers is formulated (Hierarchical-Att-GRU+Dropout).



	GRU	Hierarchical-Att-GRU	Hierarchical-Att-GRU+Dropout
MR	77.31	78.72	79.97
SUBJ	90.03	91.08	91.55
IMDB	85.78	87.61	87.92

Figure 6. Performance of the GRU-based models on the SUBJ, MR and IMDB datasets.

Both experiments showcase quite interesting findings. The experimental results on the three datasets indicate that the attention-based models achieve very promising performance as they outperform the baseline neural networks that do not make use of an attention mechanism, to a great degree in all experiments. As a result, the examined self-attention, global-attention and hierarchical-attention models lead to a performance increase of up to 3.5%.

6. Discussion

The experimental study shed light on the performance of attention-based models. In the first experiment, the focus is aimed at self-attention and global-attention methods. The results show that baseline methods like CNNs and LSTMs without the use of attention mechanisms report the lowest performance on all three datasets. Both attention mechanisms improve the performance of baseline models by a fair percentage. The use of global-attention mechanisms in LSTM models increases the performance of LSTM in all experiments when compared to the baseline methods. The use of a self-attention mechanism resulted in even better results and the experimental study indicates that self-attention LSTMs outperformed the global-attention LSTMs in all experiments. The global-attention LSTM model reports an accuracy of 88.29%, while the self-attention one boasts an accuracy of 89.12% on the IMDB dataset. Furthermore, the self-attention-based LSTMs with the addition of word2vec embeddings performed the best across all datasets, reaching an accuracy as high as 89.71% on the IMDb dataset.

In the second experiment, we examine the performance of the hierarchical-attention mechanism. The results are quite interesting and showcase that baseline GRU models can achieve satisfactory performance. The implementation of hierarchical-attention mechanisms in GRUs can greatly outperform baseline GRUs in all the examined datasets. Indeed, the hierarchical attention GRUs improve performance over baseline GRUs by almost 2% in terms of accuracy. Additionally, utilizing a dropout technique proves to be quite useful in improving the accuracy of attention-based GRUs even further.

Ultimately, the experiments showcase that the attention-based methods can lead to performance gains of up to 3.5% in terms of accuracy. In addition, it is worth noting that the attention mechanism achieves the highest accuracy gain on the IMDb dataset. The main reason for this is the dataset's particular high sequence length. Furthermore, self-attention reports the best performance gains as far as the examined attention mechanisms are concerned. In general, a self-attention mechanism can relate different positions of a single sequence to compute a representation of the same sequence. Thus, self-attention can achieve higher performance when compared to the global-attention model, mainly due to its capability to focus on very specific parts of a sequence.

7. Conclusions

In this work, we presented various attention-based models, including global-attention, self-attention and hierarchical-attention models. We implement and perform experiments on attention-based methods on RNNs, and in particular LSTMs and GRUs. The attention-based methods implemented in this work can be applied to any text classification task as well as other tasks and fields.

Overall, the attention mechanism consistently improves the accuracy of baseline models. From the experimental results, it is observed that the proposed deep neural networks utilizing attention mechanisms achieve very high performance, thus proving that they are potent and suitable tools for textual sentiment analysis as well as classification tasks in general. The classification performance that was achieved showcases the prowess and scalability of the proposed approach in analyzing users' opinions and attitudes. Furthermore, the approach that achieved the highest accuracy across the board was an LSTM on top of pre-trained 300-dimensional word2vec embeddings, which are further refined through additional training. The experimental study revealed various rather interesting findings. The best accuracy result was reported by the self-attention-based LSTM with the addition of word2vec embeddings, consolidating the fact that the examined attention-based models lead to a performance gain of up to 3.5%.

It is apparent that achieving performance gains in classifying sentiment and emotions is vital in natural language processing and particularly useful in a variety of systems and applications. Recognizing emotions, sentiments and attitudes in textual data can greatly assist in understanding peoples' opinions and is highly sought after by businesses and service providers. As a result, the attention mechanism has shown to be a strong performer, which can satisfy the aforementioned needs. A direction of future work could be to explore sequences of very high length and examine the operation of local attention or other attention mechanisms on such sequences. In addition, another direction for future work concerns the examination of self-attention, global-attention and hierarchical-attention mechanisms under alternative deep learning architectures and training parameters, such as different dropout rates. This constitutes the main direction that future work could focus on.

Author Contributions: Conceptualization, I.P. and S.K.; methodology, I.P. and S.K.; software, I.P. and S.K.; validation, I.P., S.K. and F.G.; formal analysis, S.K. and F.G.; writing—original draft preparation, I.P., S.K. and F.G.; writing—review and editing, I.P., S.K. and F.G.; supervision, I.P. and I.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are openly available in [48–50].

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

References

1. Socher, R.; Lin, C.C.; Manning, C.; Ng, A.Y. Parsing natural scenes and natural language with recursive neural networks. In Proceedings of the 28th International Conference on Machine Learning (ICML-11), Bellevue, WA, USA, 28 June–2 July 2011; Omnipress: Madison, WI, USA, 2011; pp. 129–136.
2. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
3. Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint* **2014**, arXiv:1406.1078.
4. Kamath, U.; Liu, J.; Whitaker, J. *Deep Learning for NLP and Speech Recognition*; Springer: Cham, Switzerland, 2019; Volume 84.
5. Kwon, S. A CNN-assisted enhanced audio signal processing for speech emotion recognition. *Sensors* **2020**, *20*, 183.
6. Sajjad, M.; Kwon, S. Clustering-based speech emotion recognition by incorporating learned features and deep BiLSTM. *IEEE Access* **2020**, *8*, 79861–79875.
7. Paszkiel, S. Using neural networks for classification of the changes in the EEG signal based on facial expressions. In *Analysis and Classification of EEG Signals for Brain–Computer Interfaces*; Springer: Cham, Switzerland, 2020; pp. 41–69.
8. Aladem, M.; Rawashdeh, S.A. A single-stream segmentation and depth prediction CNN for autonomous driving. *IEEE Intell. Syst.* **2020**. [CrossRef]
9. Voulodimos, A.; Doulamis, N.; Doulamis, A.; Protopapadakis, E. Deep learning for computer vision: A brief review. *Comput. Intell. Neurosci.* **2018**, *2018*. [CrossRef] [PubMed]
10. Giannopoulos, P.; Perikos, I.; Hatzilygeroudis, I. Deep learning approaches for facial emotion recognition: A case study on FER-2013. In *Advances in Hybridization of Intelligent Methods*; Springer: Cham, Switzerland, 2018; pp. 1–16.
11. Kwon, S. Att-Net: Enhanced emotion recognition system using lightweight self-attention module. *Appl. Soft Comput.* **2021**, *102*, 107101.
12. Hossain, M.S.; Muhammad, G. Emotion recognition using deep learning approach from audio–visual emotional big data. *Inf. Fusion* **2019**, *49*, 69–78. [CrossRef]
13. Anvarjon, T.; Kwon, S. Deep-net: A lightweight CNN-based speech emotion recognition system using deep frequency features. *Sensors* **2020**, *20*, 5212. [CrossRef] [PubMed]
14. Li, Y.; Shen, L. Skin lesion analysis towards melanoma detection using deep learning network. *Sensors* **2018**, *18*, 556. [CrossRef] [PubMed]
15. Liu, J.; Chang, W.C.; Wu, Y.; Yang, Y. Deep learning for extreme multi-label text classification. In Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Tokyo, Japan, 7–11 August 2017; 2017; pp. 115–124.
16. Yadav, A.; Vishwakarma, D.K. Sentiment analysis using deep learning architectures: A review. *Artif. Intell. Rev.* **2020**, *53*, 4335–4385. [CrossRef]
17. Wei, J.; Liao, J.; Yang, Z.; Wang, S.; Zhao, Q. BiLSTM with multi-polarity orthogonal attention for implicit sentiment analysis. *Neurocomputing* **2020**, *383*, 165–173. [CrossRef]
18. Dang, N.C.; Moreno-García, M.N.; De la Prieta, F. Sentiment analysis based on deep learning: A comparative study. *Electronics* **2020**, *9*, 483. [CrossRef]
19. Bengio, Y. Learning deep architectures for AI. *Found. Trends Mach. Learn.* **2009**, *2*, 1–127. [CrossRef]
20. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is all you need. In *Advances in Neural Information Processing Systems*; 2017; pp. 5998–6008. Available online: https://arxiv.org/pdf/1706.03762.pdf (accessed on 23 April 2021).
21. Basiri, M.E.; Nemati, S.; Abdar, M.; Cambria, E.; Acharya, U.R. ABCDM: An attention-based bidirectional CNN-RNN deep model for sentiment analysis. *Future Gener. Comput. Syst.* **2021**, *115*, 279–294. [CrossRef]
22. Cheng, J.; Dong, L.; Lapata, M. Long short-term memory-networks for machine reading. *arXiv* **2016**, arXiv:1601.06733.
23. Bahdanau, D.; Cho, K.; Bengio, Y. Neural machine translation by jointly learning to align and translate. *arXiv preprint* **2014**, arXiv:1409.0473.
24. 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, TX, USA, 1–5 November 2016; pp. 606–615.
25. Liu, B. *Sentiment Analysis: Mining Opinions, Sentiments, and Emotions*; Cambridge University Press, 2015; Available online: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.359.6341&rep=rep1&type=pdf (accessed on 23 April 2021).
26. Perikos, I.; Kardakis, S.; Paraskevas, M.; Hatzilygeroudis, I. Hidden Markov Models for Sentiment Analysis in Social Media. In Proceedings of the 2019 IEEE International Conference on Big Data, Cloud Computing, Data Science & Engineering (BCD), Honolulu, HI, USA, 29–31 May 2019; pp. 130–135.
27. Cai, Y.; Huang, Q.; Lin, Z.; Xu, J.; Chen, Z.; Li, Q. Recurrent neural network with pooling operation and attention mechanism for sentiment analysis: A multi-task learning approach. *Knowl. Based Syst.* **2020**, *203*, 105856. [CrossRef]
28. Munkhdalai, T.; Yu, H. Neural tree indexers for text understanding. In Proceedings of the Association for Computational Linguistics, Vancouver, Canada, 30 July–4 August 2017; Volume 1, p. 11.
29. 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.

30. Yin, Y.; Song, Y.; Zhang, M. Document-level multi-aspect sentiment classification as machine comprehension. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, 7–11 September 2017; pp. 2044–2054.
31. Lee, G.; Jeong, J.; Seo, S.; Kim, C.; Kang, P. Sentiment classification with word localization based on weakly supervised learning with a convolutional neural network. *Knowl. Based Syst.* **2018**, *152*, 70–82. [CrossRef]
32. Lin, Z.; Feng, M.; Santos CN, D.; Yu, M.; Xiang, B.; Zhou, B.; Bengio, Y. A structured self-attentive sentence embedding. *arXiv preprint* **2017**, arXiv:1703.03130.
33. Chen, H.; Sun, M.; Tu, C.; Lin, Y.; Liu, Z. Neural sentiment classification with user and product attention. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, TX, USA, 21 September 2016; pp. 1650–1659.
34. Liu, G.; Guo, J. Bidirectional LSTM with attention mechanism and convolutional layer for text classification. *Neurocomputing* **2019**, *337*, 325–338. [CrossRef]
35. Fu, X.; Yang, J.; Li, J.; Fang, M.; Wang, H. Lexicon-enhanced LSTM with attention for general sentiment analysis. *IEEE Access* **2018**, *6*, 71884–71891. [CrossRef]
36. Dou, Z.Y. Capturing user and product Information for document level sentiment analysis with deep memory network. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, 3 March 2017; pp. 521–526.
37. Li, Z.; Zhang, Y.; Wei, Y.; Wu, Y.; Yang, Q. End-to-end adversarial memory network for cross-domain sentiment classification. In Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI 2017), Melbourne, Australia, 19–25 August 2017; Available online: https://www.semanticscholar.org/paper/End-to-End-Adversarial-Memory-Network-for-Sentiment-Li-Zhang/85031a4873fe4ddda4a0841b9169b2f164980f3d?p2df (accessed on 23 April 2021).
38. Tang, D.; Qin, B.; Liu, T. Aspect level sentiment classification with deep memory network. *arXiv preprint* **2016**, arXiv:1605.08900.
39. Shuang, K.; Ren, X.; Yang, Q.; Li, R.; Loo, J. AELA-DLSTMs: Attention-Enabled and Location-Aware Double LSTMs for aspect-level sentiment classification. *Neurocomputing* **2019**, *334*, 25–34. [CrossRef]
40. Chen, J.; Hu, Y.; Liu, J.; Xiao, Y.; Jiang, H. Deep short text classification with knowledge powered attention. In Proceedings of the AAAI Conference on Artificial Intelligence 2019, Honolulu, Hawaii, 27 January–1 February 2019; Volume 33, pp. 6252–6259.
41. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient estimation of word representations in vector space. *arXiv* **2013**, arXiv:1301.3781.
42. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
43. Duchi, J.; Hazan, E.; Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.* **2011**, *12*, 2121–2159.
44. Hochreiter, S. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *Int. J. Uncertain. Fuzziness Knowl. Based Syst.* **1998**, *6*, 107–116. [CrossRef]
45. Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*; 2014; pp. 3104–3112. Available online: https://arxiv.org/pdf/1409.3215.pdf (accessed on 23 April 2021).
46. Luong, M.T.; Pham, H.; Manning, C.D. Effective approaches to attention-based neural machine translation. *arXiv preprint* **2015**, arXiv:1508.04025.
47. Socher, R.; Perelygin, A.; Wu, J.; Chuang, J.; Manning, C.D.; Ng, A.; Potts, C. Recursive deep models for semantic compositionality over a sentiment treebank. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, Seattle, DC, USA, 18–21 October 2013; pp. 1631–1642.
48. Pang, B.; Lee, L. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, Ann Arbor, MI, USA, 25–30 June 2005; pp. 115–124.
49. Pang, B.; Lee, L. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In Proceedings of the 42nd annual meeting on Association for Computational Linguistics, Barcelona, Spain, 21–26 July 2004; p. 271.
50. Maas, A.L.; Daly, R.E.; Pham, P.T.; Huang, D.; Ng, A.Y.; Potts, C. Learning word vectors for sentiment analysis. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1, Portland, OR, USA, 19–24 June 2011; pp. 142–150.
51. Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Kudlur, M.; et al. Tensorflow: A system for large-scale machine learning. In Proceedings of the 12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16), Savannah, GA, USA, 2–4 November 2016; pp. 265–283.