applied sciences

MDPI

Article

Self-Interaction Attention Mechanism Based Text Representation for Document Classification

Jianming Zheng †, Fei Cai †,*, Taihua Shao † and Honghui Chen

Science and Technology on Information Systems Engineering Laboratory,
National University of Defense Technology, Changsha 410073, China; zhengjianming12@nudt.edu.cn (J.Z.);
fengchong16@nudt.edu.cn (T.S.); chenhonghui@nudt.edu.cn (H.C.)
* Correspondence: caifei@nudt.edu.cn; Tel.: +86-155-7512-7366
† Co-first authors of this article.

Received: 17 March 2018; Accepted: 10 April 2018; Published: 12 April 2018

Abstract: Document classification has a broad application in the field of sentiment classification, document ranking and topic labeling, etc. Previous neural network-based work has mainly focused on investigating a so-called forward implication, i.e., the preceding text segments are taken as the context of the following text segments when generating the text representation. Such a scenario typically ignores the fact that the semantics of a document are a product of the mutual implication of all text segments in a document. Thus, in this paper, we introduce a concept of interaction and propose a text representation model with Self-interaction Attention Mechanism (TextSAM) for document classification. In particular, we design three aggregated strategies to integrate the interaction into a hierarchical architecture for document classification, i.e., averaging the interaction, maximizing the interaction and adding one more attention layer on the interaction, which leads to three models, i.e., TextSAM$_{AVE}$, TextSAM$_{MAX}$ and TextSAM$_{ATT}$, respectively. Our comprehensive experimental results on two public datasets, i.e., Yelp 2016 and Amazon Reviews (Electronics), show that our proposals can significantly outperform the state-of-the-art neural-based baselines for document classification, presenting a general improvement in terms of accuracy ranging from 5.97% to 14.05% against the best baseline. Furthermore, we find that our proposals with a self-interaction attention mechanism can obviously alleviate the impact brought by the increase of sentence number as the relative improvement of our proposals against the baselines are enlarged when the sentence number increases.

Keywords: interaction representation; attention mechanism; document classification; hierarchical architecture

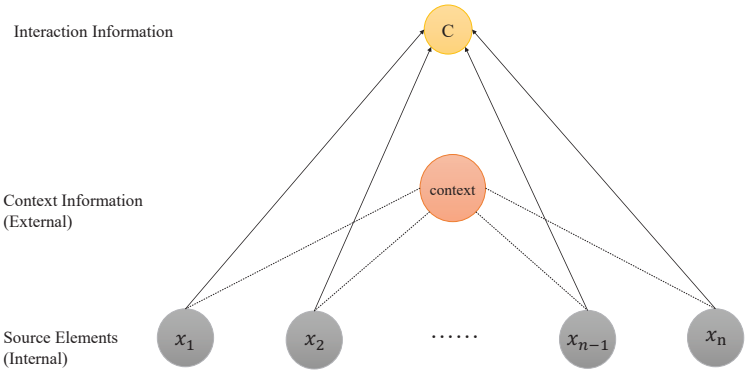
1. Introduction

Document classification, as a challenging task in the field of Natural Language Processing (NLP), typically assigns one or more class labels to a document according to its subject or other attributes, e.g., author and topic. Generally, it has a broad application in the area of sentiment classification [1,2], document ranking [3], genre classification [4] and topic labeling [5], etc.

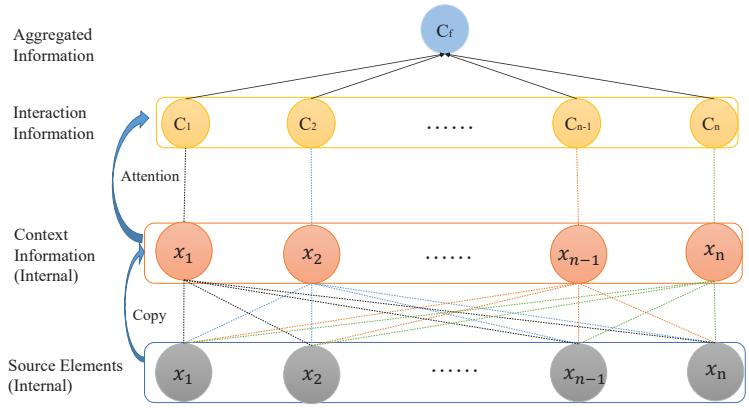
Traditional approaches on document classification mainly label the document according to a relevance of the document to a class label, which is estimated based on the statistical indicators, e.g., the frequency of co-occurrence words (the bag-of-words model [6]), the frequency of the word pair (the n-grams model [7]) and the weight scores of each word in different documents (the TF-IDF model [8]). However, such statistical-based methods typically suffer from the problem of data sparsity and dimensionality explosion when they are applied to a large-scale corpus. To deal with this, neural-based approaches are proposed by learning distributed representation [9–12]. The neural-based models generally follow a so-called one-way action. That is to say, representations generated for

the preceding text segments are taken as the context to determine the representations of following text segments. Instead, we argue that the semantics of a text segment is a product of interactions of all text segments in a document. Restrictions to one-way action may result in a partial semantic loss. Although these interaction relations may be learned by neural networks with enough samples, modeling such interaction relations can directly make document representation more informative and effective. In order to learn such interaction, we propose a text representation model with Self-interaction Attention Mechanism (TextSAM)-based text representation for document classification.

We illustrate our Self-interaction Attention Mechanism in Figure 1b. The idea of an *action* of a text segment on another segment is that the former assigns a semantic weight to the latter. Standard attention mechanism [13] used in text representation (Figure 1a) typically introduces a context vector by a random initialization [14] (external input) as the *action controller* to get source elements semantics weights that make up the document representation. That is to say, standard attention mechanism is a one-way *action* between the context vector and source elements. In contrast, our self-interaction attention mechanism (Figure 1b) resorts to all source elements (without external input) in a document as the *action controllers*. All source elements that are regarded as the *action controllers* are equivalent to the interaction between source elements. In detail, we design three aggregated strategies to integrate the interaction into a hierarchical architecture for document classification, i.e., averaging the interaction, maximizing the interaction and adding one more attention layer on the interaction, which leads to three models, i.e., TextSAM$_{AVE}$, TextSAM$_{MAX}$ and TextSAM$_{ATT}$, respectively. Our experimental results prove the effectiveness of our proposals.



(**a**) Standard Attention Mechanism



(**b**) Self-interaction Attention Mechanism

Figure 1. Comparison of Two Attention Mechanisms.

The main contributions of our work are summarized as follows:

- To the best of our knowledge, ours is the first attempt to model the interactions between source elements in a document;
- We propose a Self-interaction Attention Mechanism (TextSAM) to produce the interaction representation in a document for classification.
- We introduce three aggregated strategies to integrate the interaction into a hierarchical structure, generating three models for document classification, i.e., TextSAM$_{AVE}$, TextSAM$_{MAX}$ and TextSAM$_{ATT}$, respectively.
- We conduct comprehensive experiments on two large-scale public datasets (Yelp 2016 and Amazon Reviews (Electronics)) for document classification. We find that our proposals significantly outperform the state-of-the-art baselines, achieving an improvement ranging from 5.97% to 33.27% in terms of accuracy.

The remainder of this paper is organized as follows: we describe the related works in Section 2. Our proposals are described in Section 3. Section 4 presents our experimental setup. In Section 5, we report and discuss our experimental results. Finally, we conclude in Section 6.

2. Related work

In this section, we briefly summarize the general statistical approaches for document classification in Section 2.1 and the neural networks-based methods in Section 2.2. We then describe the recent work on attention mechanism for document classification in Section 2.3.

2.1. Statistical Classification

The most common and simplest approaches for document classification are Bag-of-Words (Bow) models [6] or n-grams models [7], which regard each word or a word pair in text as a discrete entity and employ one-hot representation to reflect the frequency of a word or word pair. However, the BoW model can only symbolize the entities and cannot reflect the semantic relationship between entities. In view of that, Zhang et al. [7] covert the one hot representation into the word2vec [15] representation. Furthermore, in order to highlight how important a word is to a document, the TF-IDF term-weighting scheme is added to improve the performance of document classification [8]. Other works incorporate the text features into model construction, e.g., the noun phrases [16] and the tree kernels [17].

Clearly, a progressive step has been made in statistical classification, but these approaches inevitably suffer from the problem of data sparsity and dimensionality explosion when they are employed in a large-scale corpus. Instead, our proposals built on neural architecture have the ability to learn the distributed representation to deal with the above drawbacks.

2.2. Neural Classification

Recently, deep learning techniques have attracted considerable attention in the field of document classification. For instance, Joulin et al. [9] utilize a hidden layer to integrate all inputs for document representation, which leads to an excellent performance for document classification. Kim [10] directly employs the convolutional neural networks (CNNs) architecture for text classification.Similarly, at the character level, Zhang et al. [7] use the CNNs architecture to represent the text. Liu et al. [11] combine the multitask learning framework with recurrent neural networks (RNNs) structure to jointly learn across multiple related tasks. Furthermore, Lai et al. [12] adopt the recurrent structure to grasp the context information and can identify the key components in the text by employing a max-pooling layer.

Although these approaches have been proved effective in the task of document classification, they completely depend on the structure of a network to implicitly represent a document, ignoring the interaction that may exist among the source elements in a document, e.g., words or sentences. However, our proposals can model the interaction as the starting point to better reflect the semantic relationship between each component in a document, which we argue can help improve the effectiveness of document classification.

2.3. Attention Mechanism

Since Bahdanau et al. [13] first proposeed the attention mechanism in the field of machine translation, the attention mechanism has become a standard part of natural language processing (NLP), e.g., neural machine translation [18], image caption [19], speech recognition [20] and question answering [21]. Standard attention mechanism is actually a process that computes a categorical distribution to make soft-selection over source elements [22]. This paradigm makes it possible to control the interaction between source elements with their surrounding context in a document.

Generally, such attention mechanism-based approaches incorporate the context either by a random initialization or by a joint learning process [14]. In addition, they are not directly applicable to the tasks like sentiment classification that has only one single sentence as input [23]. However, our proposal based on the self-interaction attention mechanism can use each source element as context without extra input, which helps to develop the potential of interaction in the attention mechanism.

3. Methods

In this section, we first formally describe our proposal, i.e., the self-interaction attention mechanism, in Section 3.1. After that, we introduce three aggregated strategies to integrate the interaction into the hierarchical architecture in Section 3.2. Finally, we describe the process of classification in Section 3.3.

3.1. Self-Interaction Attention Mechanism

3.1.1. One-Way Action

As shown in Figure 1a, the standard attention mechanism is implemented as a hidden layer which carries a soft-selection process over the source elements. The context vector, as the *action controller* of the one-way action, assigns source elements the semantics weights that form the document representation.

Formally, we can define a sequence of source elements as $x = \{x_1, x_2, \cdots, x_n\}$, where x_i $(1 \leq i \leq n)$ is an input source element that is vectorized by a representation h_i through RNN. We first feed the input vector representation h_i through a one-layer Multi Layer Perception (MLP) to get u_i as a hidden representation of h_i, i.e.,

$$u_i = tanh(W_h h_i + b_h),$$

where W_h and b_h are a weight matrix and a bias term, respectively. Similarly, we can have a hidden representation u_w of the context vector. Then, we formulate the one-way action representation c between the source elements of a document and context vector as follows:

$$c = \sum_{i=1}^{n} \alpha_i h_i,$$

where

$$\alpha_i = \frac{exp(u_i^T u_w)}{\sum_{i=1}^{n} exp(u_i^T u_w)}$$

is the semantics weight of the source element x_i assigned by the *action controller*, i.e., context vector u_w.

3.1.2. Interaction Representation

Clearly, this attention mechanism can tackle the issue of compressing a source element with variable-length memory into a fixed-dimensional vector. However, the context vector u_w is usually an external input, which is not applicable to some tasks. Accordingly, in this paper, we take each source element as context to formulate a deep one-way action without an external information input. Therefore, as shown in Figure 2, with the self-interaction attention mechanism, we can similarly

represent the one-way action c_k between all source elements and the source element x_k (as *action controller*) as follows:

$$c_k = \sum_{i=1}^{n} \alpha_{ki} h_i,$$

where

$$\alpha_{ki} = \frac{exp(u_i^T u_k)}{\sum_{i=1}^{n} exp(u_i^T u_k)}$$

is the semantics weight of the source element x_i assigned by the *action controller*, i.e., source element x_k.

Enumerating all source elements as the *action controller*, we can get the one-way action sequence $(c_1, c_2, \cdots, c_n)$, which is equivalent to realizing the interaction between source elements. For simplicity, we denote an interaction representation C as follows:

$$C = (c_1, c_2, \cdots, c_n).$$

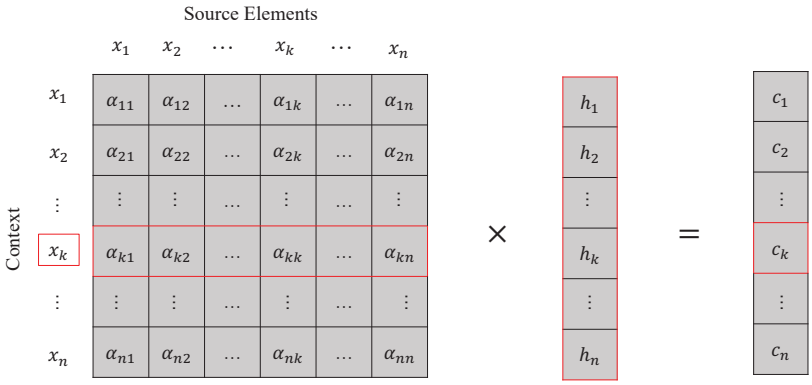


Figure 2. Process of Self-interaction Attention Mechanism.

3.2. Aggregated Strategy

After illustrating the self-interaction attention mechanism, we should convert the variable-length interaction representation C into a fixed-dimensional text representation t. Hence, we propose three aggregated strategies, i.e., averaging the interaction, maximizing the interaction as well as adding one more attention layer on the interaction, which results in TextSAM$_{AVE}$, TextSAM$_{MAX}$ and TextSAM$_{ATT}$, respectively.

3.2.1. Pooling Proces

In order to convert the variable-length input into a fixed-length representation, we perform a pooling operation along the first dimension of interaction representation C (as shown in Figure 3) and particularly introduce two strategies for pooling, i.e., averaging the interaction and maximizing the interaction, resulting in TextSAM$_{AVE}$ and TextSAM$_{MAX}$, respectively.

TextSAM$_{AVE}$ assumes that each one-way action c_i in C is equal to the final document representation. Therefore, TextSAM$_{AVE}$ employs the average pooling [24] in the pooling layer as

$$t = \frac{1}{n} \sum_{i=1}^{n} c_i.$$

TextSAM$_{MAX}$ focuses on extracting the most important feature from the interaction representation C. Thus, TextSAM$_{MAX}$ employs the max pooling [25] in the pooling layer as

$$t = max\{c_i\},$$

where $max\{\cdot\}$ means to get the maximum value in each dimension of the interaction c_i $(i = 1, \cdots, n)$.

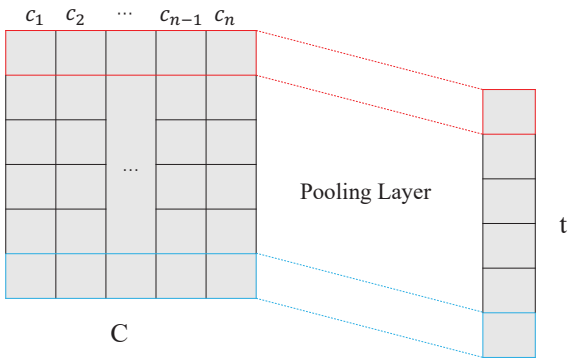


Figure 3. Pooling Process in the Aggregated Strategy.

3.2.2. One-Way Action-Again Process

As shown in Section 3.1, we can find that the standard attention mechanism has the ability to integrate the variable-length input into a fixed-dimensional representation. In addition, as the interaction can not contribute equally to the final text representation, we add another layer of attention in the aggregated strategy to develop a deep one-way action.

As shown in Figure 4, we first feed the interaction c_i through a one-layer MLP to get the corresponding hidden representation v_i, i.e.,

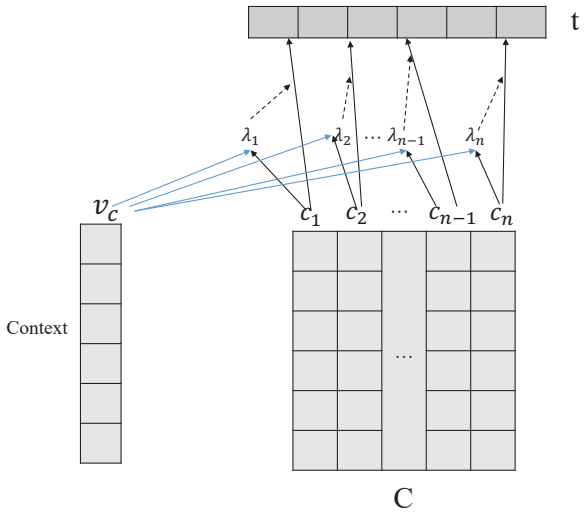


Figure 4. Interaction-Again Process in the Aggregated Strategy.

$$v_i = tanh(W_c c_i + b_c),$$

where W_c and b_c are a weight matrix and a bias term, respectively. Then, we randomly initialize the context vector v_c and employ a softmax function as follows:

$$\lambda_i = \frac{v_i^T v_c}{\sum v_i^T v_c},$$

where λ_i is the semantics weight assigned by the *action controller*, i.e., context vector v_c. Finally, the document representation t can be represented as:

$$t = \sum_{i=1}^{n} \lambda_i c_i.$$

So far, our proposals with a self-interaction attention mechanism have been illustrated completely.

3.3. Document Classification

In the process of class prediction, we apply a softmax classifier on the document representation t to get a predicted label $\hat{t}$, where $\hat{t} \in \mathcal{Y}$ and $\mathcal{Y}$ is the class label set, i.e.,

$$\hat{t} = \text{argmax}\, p(\mathcal{Y}|t),$$

where

$$p(\mathcal{Y}|t) = \text{softmax}(W^{(t)}t + b^{(t)}).$$

Here, $W^{(t)}$ and $b^{(t)}$ are the reshape matrix and the bias term, respectively.

Therefore, we can use the negative log-likelihood to define the loss function L as follows:

$$L = -\log p(\hat{t}|t).$$

4. Experiments

In this section, we first present our proposed models and baseline methods in Section 4.1. We then describe the evaluation metrics and datasets used in our experiments in Section 4.2. Next, we describe our model setup in Section 4.3 in detail and list the research questions to be answered in Section 4.4 that guide our experiments.

4.1. Model Summary

Since the hierarchical architecture has been proved effective in the field of document classification [14], we adopt the hierarchical architecture in our models, i.e., the word level and the sentence level.

In addition, in the standard attention mechanism, it only calculates a one-way action, i.e., the complexity for it is $O(n)$. In contrast, a self-interaction attention mechanism should calculate n one-way actions, i.e., the complexity for it is $O(n^2)$. Hence, in order to avoid the problem of extremely high complexity, we adopt the standard attention mechanism at the word level and the self-interaction attention mechanism at the sentence level. The detailed process of our proposals is illustrated in Algorithm A1 (see Appendix A).

For comparison, we summarize our proposed models and the baseline methods in Table 1.

Table 1. Models summary. (The baseline methods and our proposals are identified with the identifiers ∘ and ⋆, respectively.)

Models	Description
TextRNN ∘	A recurrent neural network-based approach [11].
TextHAN ∘	A hierarchical attention network-based approach [14].
TextSAM$_{AVE}$ ⋆	A self-interaction attention mechanism-based approach with averaging the interaction.
TextSAM$_{MAX}$ ⋆	A self-interaction attention mechanism-based approach with maximizing the interaction.
TextSAM$_{ATT}$ ⋆	A self-interaction attention mechanism-based approach with one more attention on interaction.

4.2. Datasets and Evaluation Metrics

We implement our experiments on two large scale public datasets that can be used for document classification, i.e., Yelp 2016 and Amazon Reviews (Electronics). The statistics of the datasets are summarized in Table 2.

Table 2. Statistics of datasets (The vocabulary in datasets has gone through data cleaning, excluding single characters and punctuations as well as retaining only the lemmatized words).

Dataset	Yelp 2016	Amazon Reviews (Electronics)
# classes	5	5
# documents	4,153,150	1,689,188
# average sentences/document	8.11	6.88
# average words/sentence	17.02	7.65
# average words/document	138.02	136.97
# maximal sentences in document	166	416
# maximal words in document	1431	7488
# words in vocabulary	155,498	66,551

- **Yelp 2016** is obtained from the Yelp Dataset Challenge in 2016 (https://www.yelp.com/dataset/challenge), which has five levels of ratings from 1 to 5. In other words, we can classify the documents into five classes.
- **Amazon Reviews (Electronics)** are obtained from Amazon products data (http://jmcauley.ucsd.edu/data/amazon/). This dataset contains the product reviews and the metadata from Amazon from May 1996 to July 2014. Similarly, five levels of ratings from 1 to 5 are given to product reviews.

As shown in Table 2, the most notable differences between Yelp 2016 and Amazon Reviews (Electronics) lie in the number of documents and the size of vocabulary, which could have an impact on the performance of document classification.

For evaluation, we use *accuracy* as the metric, which is a standard metric to measure the overall document classification performance. In detail, the metric *accuracy* can be computed as

$$accuracy = \frac{\sum_{i=1}^{k} Sgn(predict(i), ground\_truth(i))}{k},$$

where k is the total number of test documents, $Sgn(a, b)$ is a sign function ($Sgn(a, b) = 1$ when a equals b; otherwise, $Sgn(a, b) = 0$.), $ground\_truth(i)$ indicates the ground truth of the class label for document i and $predict(i)$ returns the predicted class label for document i.

4.3. Model Configuration

For data processing, in order to construct the hierarchical architecture, we split the documents into sentences and tokenize each sentence using the Stanford's CoreNLP [26]. Besides, in order to avoid the problem of vocabulary redundancy, we discard the words with a single character or with punctuations. Finally, the top 100,000 words in Yelp 2016 and 50,000 words in Amazon Reviews (Electronics) remained the same. In addition, we use the pre-trained word vectors dataset *GloVe* (Wikipedia 2014 + Gigaword 5) (http://opendatacommons.org/licenses/pddl/) as our embedding dataset.

For the model setup, we give the final setting as follows. the batch size is set to be 64, i.e., 64 documents per batch, the word embedding dimension is set to be 200 and the LSTM dimension is set to 50. In training process, we use the stochastic gradient descent approach to train all models with a learning rate 0.001. To avoid the gradient problem, we adopt a gradient clipping [27] by scaling gradients when the norm exceeds a threshold of 5. In addition, as shown in Table 2, we see that $\#averagesentences/document$ and $\#averagewords/sentence$ are both <30. Hence, we set the truncation number of the sentence to 30.

For initializing the neural networks, we adopt the xavier initialization approach to keep the scale of the gradients roughly the same in all layers [28].

4.4. Research Question

The research questions guiding our experiments are listed as follows.

RQ 1 Does the self-interaction attention mechanism incorporated in the document representation model help to improve the performance for classification?

RQ 2 Since the self-interaction attention mechanism is only built on the sentence level, what is the impact on performance of the number of sentences in a document?

Answers to these two questions would provide valuable insights into the utility of self-interaction attention in neural network-based models for document classification.

5. Results and Analysis

In Section 5.1, we compare the performance of our proposals against that of the baseline methods on two datasets. Then, Section 5.2 zooms in on the effect on document classification by the number of sentences in a document. In addition, we analyze the impact on performance of the truncation number in Section 5.3.

5.1. Performance Comparison

To answer **RQ1**, we adopt the holdout method (see Section 5.1.1) and the k-fold cross-validation (see Section 5.1.2) to compare the classification performance, respectively.

5.1.1. Holdout Method

In the holdout method, for each dataset, we randomly divide the data into 10 equally sized subsamples. Then we select 8 subsamples for training, a single subsample for validation and a single subsample for testing. In Table 3, we present the experimental results of all discussed models in this paper for document classification on Yelp 2016 and Amazon Reviews (Electronics), respectively.

Table 3. Accuracy on the document classification task in the holdout methods. The results of the best baseline and the best performer in each column are underlined and boldfaced, respectively.

Model	Yelp 2016	Amazon Reviews (Electronics)
TextRNN [11]	0.4433	0.5127
TextHAN [14]	<u>0.5575</u>	<u>0.5493</u>
TextSAM$_{AVE}$	0.5507	0.5636
TextSAM$_{MAX}$	**0.5908**	**0.6265**
TextSAM$_{ATT}$	0.5587	0.5709

Regarding the baselines, TextHAN outperforms TextRNN, showing an improvement of 25.76% and 7.14% in terms of accuracy in Yelp 2016 and Amazon Reviews(Electronics), respectively. This means that hierarchical architecture does indeed represent the document well for classification. In addition, our proposal with the self-interaction mechanism, i.e., TextSAM$_{AVE}$, TextSAM$_{MAX}$ and TextSAM$_{ATT}$, generally outperform the baseline model (except that TextSAM$_{AVE}$ loses the competition with TextHAN on Yelp 2016.). This proves that our proposed self-interaction attention mechanism can significantly promote the performance of document classification.

In particular, TextSAM$_{MAX}$ is the best performing model among our proposals. On the Yelp 2016 dataset, TextSAM$_{MAX}$ shows an obvious improvement of 5.97% against the best baseline, TextHAN, and achieves an improvement of 7.28% against TextSAM$_{AVE}$ and of 5.75% against TextSAM$_{ATT}$. Similarly, on Amazon Reviews (Electronics), compared with TextHAN, TextSAM$_{AVE}$ and TextSAM$_{ATT}$,

TextSAM$_{MAX}$ achieves improvements of 14.05%, 11.56% and 9.74%. By applying the max pooling operation on each dimension of interaction representation, TextSAM$_{MAX}$ can extract the most representative feature to better represent a document.

TextSAM$_{ATT}$, like TextSAM$_{MAX}$, outperforms TextHAN by 0.22% in terms of accuracy on Yelp 2016 and 3.93% on Amazon Reviews (Electronics). TextSAM$_{AVE}$ performs worse than TextSAM$_{ATT}$ but still improves the accuracy of 2.60% over TextHAN on Amazon Reviews (Electronics). The difference in performance between TextSAM$_{ATT}$ and TextSAM$_{AVE}$ may be explained by the fact that averaging the interaction will ignore the fact that each document has its own emphasis and specific topic.

5.1.2. K-Fold Cross-Validation

In the k-fold cross-validation, we first randomly divide the dataset into 5 equally sized subsets. We select four subsets for training and the remaining one for testing. The cross-validation process is then repeated 5 times to make sure that each subset can be used as the testing data. We report the average accuracy from these five experiments as the final classification in Table 4.

Table 4. Accuracy on the document classification task in the k-fold cross-validation. The results of the best baseline and the best performer in each column are underlined and boldfaced, respectively.

Model	Yelp 2016	Amazon Reviews (Electronics)
TextRNN [11]	0.4532	0.5211
TextHAN [14]	<u>0.5537</u>	<u>0.5435</u>
TextSAM$_{AVE}$	0.5543	0.5632
TextSAM$_{MAX}$	**0.5919**	**0.6293**
TextSAM$_{ATT}$	0.5602	0.5726

The results in Table 4 are similar to those in Table 3. We conclude that when the number of training samples is large enough, the holdout method can be used for experimental evaluation instead of the k-fold cross-validation.

5.2. Impact of the Number of Sentences

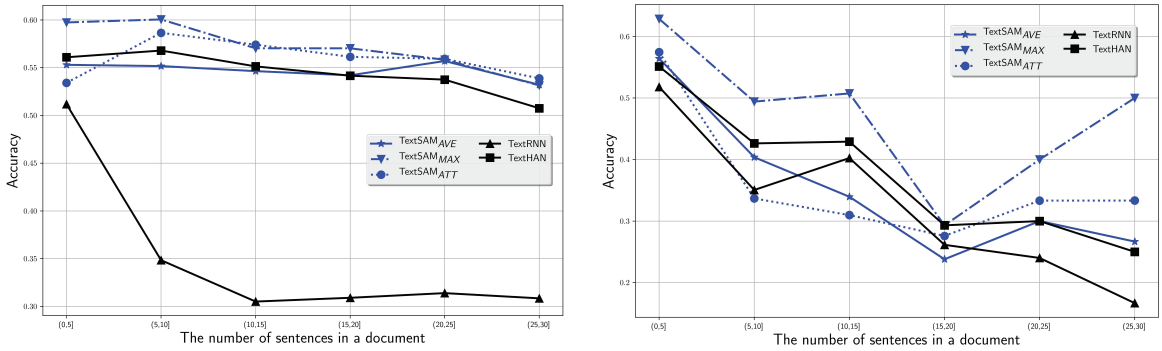
To answer **RQ 2**, we manually group the documents according to the number of sentences, e.g., $(0, 5]$, $(5, 10]$, $(10, 15]$, $(15, 20]$, $(20, 25]$ and $(25, 30]$ (the truncation number of the sentence is 30), and then examine the performance of our proposals as well as the baselines on groups of documents with various sentence numbers. We plot the results in Figure 5a,b on Yelp 2016 and Amazon Reviews (Electronics), respectively.

Clearly, for both datasets, we find that the performance of all discussed models declines monotonously as the sentence number increases. The higher the sentence number, the more complex the relation between sentences in the document, making it more difficult to get a good document representation.

On the Yelp 2016 dataset, in particular, when the number of sentences increases from $(0, 5]$ to $(25, 30]$, the accuracy of all discussed models presents a significant drop. For instance, regarding the baseline methods, TextRNN and TextHAN decreases by around 20% and 6% in terms of accuracy, respectively. Regarding our proposals with self-interaction attention mechanism, generally, the TextSAM models present a relatively stable decrease in terms of accuracy when the number of sentences increases. For instance, the drop rate of TextSAM$_{AVE}$ achieves at most 5% when performing on documents with sentence number from $(0, 5]$ to $(25, 30]$. In addition, our proposals consistently outperform the baseline models when the number of sentences exceeds $(15, 20]$. On the Amazon Reviews (Electronics) dataset, similar results can be found. In general, the baseline models present a stable decline as the number of sentences increases. However, our proposals show an obvious drop in terms of accuracy before the number of sentences reaches $(15, 20]$. After that, different from the

stable descending trend on Yelp 2016, the performance of TextSAM models consistently jumps until the number of sentences arrives at $(25, 30]$.

From the above observation, we would like to conclude that, compared with the baseline models, our proposals can obviously alleviate the impact brought about by the increase of sentence number on the performance of document classification. Since the baseline models are typically based on the LSTM (Long Short-Term Memory) architecture, which suffers from the problem of gradient vanishing [29] and make a descending performance as the number of sentences increases. Instead, our proposals can tackle such a problem by introducing the interaction between source elements and the context into the hierarchical architecture, which leads us to retain the overall semantics of text and help improve the performance of document classification.

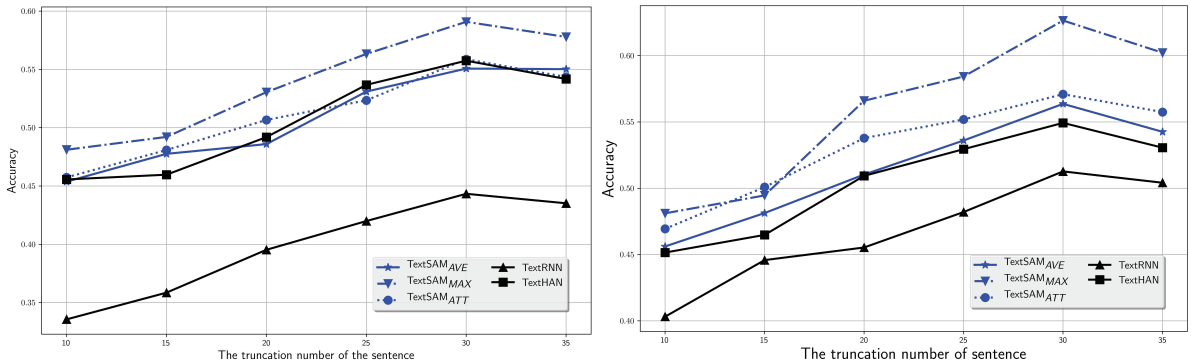


(**a**) Yelp 2016 (**b**) Amazon Reviews (Electronics)

Figure 5. Classification accuracy on documents with various number of sentences.

5.3. Parameters Analysis

Since the truncation number in our experimental setup (see Section 4.3) is artificially fixed, we undertake a further investigation on the performance of our proposals with a different truncation number, e.g., $10, 15, \cdots, 35$. We plot the results in Figure 6a,b on Yelp 2016 and Amazon Reviews (Electronics), respectively.



(**a**) Yelp 2016 (**b**) Amazon Reviews (Electronics)

Figure 6. Classification accuracy on documents with various truncation number of the sentence.

Clearly, on Yelp 2016, as the truncation number increases, all discussed models increase at first, and then decrease. In particular, all discussed models reach a peak at 30. Similar findings can be found in Amazon Reviews (Electronics). These findings indicate that our hypothesis that the truncation number is set to 30 is reasonable.

6. Conclusions

In this paper, we introduce a concept called interaction in document representation and then design a self-interaction attention mechanism to inject the interaction into a hierarchical architecture for document classification. In particular, based on a hierarchical architecture, we propose three strategies to integrate the interaction for document representation, i.e., TextSAM$_{AVE}$, TextSAM$_{MAX}$ and TextSAM$_{ATT}$, corresponding to averaging the interaction, maximizing the interaction and adding one more attention on the interaction, respectively.

Our experimental results on two public datasets, i.e., Yelp 2016 and Amazon Reviews (Electronics), demonstrate that our proposals significantly outperform the baseline models, i.e., TextRNN [11] and TextHAN [14]. Among of our newly proposed models, TextSAM$_{MAX}$ is superior to the other proposed models. In detail, TextSAM$_{MAX}$ presents an improvement ranging from 5.97% to 14.05% against the best baseline, i.e., TextHAN. Furthermore, we conclude that our proposals combined with the self-interaction attention mechanism can alleviate the impact brought about by the increase of sentence number.

Acknowledgments: This work was partially supported by the National Natural Science Foundation of China under No. 61702526 and the National Advanced Research Project under No. 6141B0801010b.

Author Contributions: Jianming Zheng has made substantial contribution to the design of the work, the acquisition, analysis and the interpretation of data for the work; Fei Cai drafts the work and revise it critically for important intellectual content; Taihua Shao finishes the final version to be published; Honghui Chen make an agreement to be accountable for all aspects of the work in ensuring that integrity of any part of the work are appropriately investigated and resolved.

Conflicts of Interest: The founding sponsors had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, and in the decision to publish the results.

Appendix A

Algorithm A1 Self-interaction attention mechanism for document classification

Input: The embedding matrix for each word in vocabulary, W_e; the sentence sequence in document d, $d = \{s_1, s_2, \cdots, s_{n_2}\}$; the word sequence in each sentence, e.g., $s_i = \{w_{s_i}^1, w_{s_i}^2, \cdots, w_{s_i}^{n_3}\}$.

Output: The class label of document d.

Initialize the network N_s in sentence-level and N_w in word-level. Initialize the context vector u_w in word level. Initialize all weight matrix W and bias term b.
$i = 0$
while $i < n_2$ **do**
  $j = 0$
  while $j < n_3$ **do**
    Look up the embedding matrix W_e to get the embedding of word $w_{s_i}^j$:$\mathbf{w_{s_i}^j}$.
    $j = j + 1$
  end while
  Feed the input $\{\mathbf{w_{s_i}^1}, \mathbf{w_{s_i}^2}, \cdots, \mathbf{w_{s_i}^{n_3}}\}$ through the network N_w to get the output sequence $\{h_{s_i}^1, h_{s_i}^2, \cdots, h_{s_i}^{n_3}\}$.
  Feed the output sequence through MLP to get the hidden representation sequence $\{u_{s_i}^1, u_{s_i}^2, \cdots, u_{s_i}^{n_3}\}$.
  Feed the hidden representation sequence through the standard attention mechanism to get the sentence representation (*action controller* u_w): $\mathbf{s_i} = Attention(u_{s_i}^1, u_{s_i}^2, \cdots, u_{s_i}^{n_3}; u_w)$
  $i + 1$
end while
Feed the sentence sequence through MLP to get the hidden representation sequence for sentence:$\{u_1, u_2, \cdots, u_{n_3}\}$

$k = 0$
while $k < n_3$ **do**
  Regard the u_k as the *action controller* and through the standard attention mechanism to get the one-way action representation: $c_k = Attention(u_1, u_2, \cdots, u_{n_3}; u_k)$
  $k + 1$
end while
Employ specific aggregated strategy on interaction representation C to get the document representation $t : t = aggregate(c_1, c_2 \cdots, c_{n_3})$
Employ softmax classifier on document representation t, i.e., $p = softmax(W^{(t)}t + b^{(t)})$.
return The position of the max value in p

[CrossRef]

References

1. Moraes, R.; Valiati, J.A.F.; Neto, W.P.G.A. Document-level sentiment classification: An empirical comparison between SVM and ANN. *Expert Syst. Appl.* **2013**, *40*, 621–633. [CrossRef]
2. Tang, D.; Qin, B.; Liu, T. Document Modeling with Gated Recurrent Neural Network for Sentiment Classification. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, 17–21 September 2015; pp. 1422–1432.
3. Wang, M.; Liu, M.; Feng, S.; Wang, D.; Zhang, Y. A Novel Calibrated Label Ranking-based Method for Multiple Emotions Detection in Chinese Microblogs. In *Natural Language Processing and Chinese Computing*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 238–250.
4. Santini, M.; Rosso, M. Testing a genre-enabled application: A preliminary assessment. In Proceedings of the Bcs Irsg Conference on Future Directions in Information Access, London, UK, 22 September 2008; p. 7.
5. Wang, S.; Manning, C.D. Baselines and bigrams: Simple, good sentiment and topic classification. In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers, Jeju Island, Korea, 8–14 July 2012; pp. 90–94.
6. Joachims, T. Text categorization with Support Vector Machines: Learning with many relevant features. In Proceedings of the European Conference on Machine Learning, Chemnitz, Germany, 21–23 April 1998; pp. 137–142.
7. Zhang, X.; Zhao, J.; Lecun, Y. Character-level Convolutional Networks for Text Classification. In Proceedings of the Advances in Neural Information Processing Systems, Montréal, QC, Canada, 7–12 December 2015; pp. 649–657.
8. Robertson, S. Understanding inverse document frequency: On theoretical arguments for IDF. *J. Doc.* **2004**, *60*, 503–520. [CrossRef]
9. Joulin, A.; Grave, E.; Bojanowski, P.; Mikolov, T. Bag of Tricks for Efficient Text Classification. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, Valencia, Spain, 3 April 2017; pp. 427–431.
10. Kim, Y. Convolutional Neural Networks for Sentence Classification. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Doha, Qatar, 25–29 October 2014; pp. 1746–1751.
11. Liu, P.; Qiu, X.; Huang, X. Recurrent neural network for text classification with multi-task learning. In Proceedings of the International Joint Conference on Artificial Intelligence, New York, NY, USA, 9–15 July 2016; pp. 2873–2879.
12. Lai, S.; Xu, L.; Liu, K.; Jun, Z. Recurrent Convolutional Neural Networks for Text Classification. In Proceedings of the Association for the Advancement of Artificial Intelligence, Austin, TX, USA, 25–30 January 2015; pp. 2267–2273.
13. Bahdanau, D.; Cho, K.; Bengio, Y. Neural Machine Translation by Jointly Learning to Align and Translate. In Proceedings of the International Conference on Learning Representation, Rossland, BC, Canada, 7–9 May 2015.

14. Yang, Z.; Yang, D.; Dyer, C.; He, X.; Smola, A.; Hovy, E. Hierarchical Attention Networks for Document Classification. In Proceedings of the 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.
15. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient Estimation of Word Representations in Vector Space. In Proceedings of the International Conference on Learning Representations, Scottsdale, Arizona, 2–4 May 2013.
16. Lewis, D.D. An evaluation of phrasal and clustered representations on a text categorization task. In Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Copenhagen, Denmark, 21–24 June 1992; pp. 37–50.
17. Post, M.; Bergsma, S. Explicit and Implicit Syntactic Features for Text Classification. In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, Sofia, Bulgaria, 4–9 August 2013; pp. 866–872.
18. Luong, M.T.; Pham, H.; Manning, C.D. Effective Approaches to Attention-based Neural Machine Translation. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, 17–21 September 2015.
19. Xu, K.; Ba, J.; Kiros, R.; Cho, K.; Courville, A.; Salakhutdinov, R.; Zemel, R.; Bengio, Y. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. In Proceedings of the International Conference on Machine Learning, Lille, France, 6–11 July 2015.
20. Chorowski, J.; Bahdanau, D.; Serdyuk, D.; Cho, K.; Bengio, Y. Attention-Based Models for Speech Recognition. In Proceedings of the Advances in Neural Information Processing Systems, Montréal, QC, Canada, 7–12 December 2015.
21. Hermann, K.M.; Kočiský, T.; Grefenstette, E.; Espeholt, L.; Kay, W.; Suleyman, M.; Blunsom, P. Teaching machines to read and comprehend. In Proceedings of the Advances in Neural Information Processing Systems, Montréal, QC, Canada, 7–12 December 2015.
22. Kim, Y.; Denton, C.; Hoang, L.; Rush, A.M. Structured Attention Networks. In Proceedings of the International Conference on Learning Representation, Toulon, France, 24–26 April 2017.
23. Lin, Z.; Feng, M.; Santos, C.N.D.; Yu, M.; Xiang, B.; Zhou, B.; Bengio, Y. A Structured Self-attentive Sentence Embedding. In Proceedings of the International Conference on Learning Representation, Toulon, France, 24–26 April 2017.
24. Lecun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
25. Ranzato, M.A.; Boureau, Y.L.; Lecun, Y. Sparse feature learning for deep belief networks. In Proceedings of the International Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 3–6 December 2007; pp. 1185–1192.
26. Manning, C.D.; Surdeanu, M.; Bauer, J.; Finkel, J.; Bethard, S.J.; Mcclosky, D. The Stanford CoreNLP Natural Language Processing Toolkit. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, Baltimore, MD, USA, 22–27 June 2014; pp. 55–60.
27. Pascanu, R.; Mikolov, T.; Bengio, Y. On the difficulty of training Recurrent Neural Networks. *Comput. Sci.* **2012**, *52*, III–1310.
28. Glorot, X.; Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the 13th International Conference on Artificial Intelligence and Statistics, Sardinia, Italy, 13–15 May 2010; pp. 249–256.
29. Graves, A. *Long Short-Term Memory*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 1735–1780.