symmetry

Article

Research on a Capsule Network Text Classification Method with a Self-Attention Mechanism

Xiaodong Yu [1], Shun-Nain Luo [1], Yujia Wu [1], Zhufei Cai [1], Ta-Wen Kuan [1,*] and Shih-Pang Tseng [1,2,*]

[1] School of Information Science and Technology, Sanda University, Shanghai 201209, China; xdyu@sandau.edu.cn (X.Y.); snluo@sandau.edu.cn (S.-N.L.); wuyujia@sandau.edu.cn (Y.W.); f20015110@st.sandau.edu.cn (Z.C.)

[2] Changzhou College of Information Technology, Changzhou 213164, China

* Correspondence: dwguan@sandau.edu.cn (T.-W.K.); tsengshihpang@czcit.edu.cn (S.-P.T.); Tel.: +86-13690233158 (T.-W.K.); +86-13961479106 (S.-P.T.)

Abstract: Convolutional neural networks (CNNs) need to replicate feature detectors when modeling spatial information, which reduces their efficiency. The number of replicated feature detectors or labeled training data required for such methods grows exponentially with the dimensionality of the data being used. On the other hand, space-insensitive methods are difficult to encode and express effectively due to the limitation of their rich text structures. In response to the above problems, this paper proposes a capsule network (self-attention capsule network, or SA-CapsNet) with a self-attention mechanism for text classification tasks, wherein the capsule network itself, given the feature with the symmetry hint on two ends, acts as both encoder and decoder. In order to learn long-distance dependent features in sentences and encode text information more efficiently, SA-CapsNet maps the self-attention module to the feature extraction layer of the capsule network, thereby increasing its feature extraction ability and overcoming the limitations of convolutional neural networks. In addition, in this study, in order to improve the accuracy of the model, the capsule was improved by reducing its dimension and an intermediate layer was added, enabling the model to obtain more expressive instantiation features in a given sentence. Finally, experiments were carried out on three general datasets of different sizes, namely the IMDB, MPQA, and MR datasets. The accuracy of the model on these three datasets was 84.72%, 80.31%, and 75.38%, respectively. Furthermore, compared with the benchmark algorithm, the model's performance on these datasets was promising, with an increase in accuracy of 1.08%, 0.39%, and 1.43%, respectively. This study focused on reducing the parameters of the model for various applications, such as edge and mobile applications. The experimental results show that the accuracy is still not apparently decreased by the reduced parameters. The experimental results therefore verify the effective performance of the proposed SA-CapsNet model.

Keywords: natural language processing; text classification; capsule network; self-attentive mechanism

Citation: Yu, X.; Luo, S.-N.; Wu, Y.; Cai, Z.; Kuan, T.-W.; Tseng, S.-P. Research on a Capsule Network Text Classification Method with a Self-Attention Mechanism. *Symmetry* **2024**, *16*, 517. https://doi.org/10.3390/sym16050517

Academic Editor: Lorentz Jäntschi

Received: 14 February 2024
Revised: 7 April 2024
Accepted: 8 April 2024
Published: 24 April 2024

1. Introduction

A current research hotspot, deep learning methods can express the same information through denser low latitude features and have strong feature learning capabilities, having been found to achieve good performance in text classification tasks. In the field of deep learning, self-attention mechanisms [1], an emerging technology in recent years, have been widely used in various tasks, such as face recognition, machine translation, and chatbots. They have also played a huge role in the field of natural language processing, where attention mechanisms can be applied to focus on the key points of the text and discard information unrelated to the task. In recent years, scientific research has confirmed that self-attention mechanisms have good effects on various natural language processing tasks [2–4]. In this context, studying capsule network text classification based on self-attention mechanisms has significant practical value.

Symmetry provides a fundamental way to abstract the representations of various complex systems. Identifying symmetry can substantially simplify the modeling of complex physical systems [5]. The neural network, a nature-inspired computing model with some symmetrical architectures, can provide an available and effective approach to explore the symmetry in the data source [6]. In [7], the authors have demonstrated the successful application of a neural network to explore symmetry in order to estimate the velocity in blood microflows. In [8], a neural network was used to validate the facial symmetry and asymmetry of human faces. In this paper, we proposed a new capsule network with the self-attention mechanism for exploring the symmetry from the text corpus to reduce the parameters of model in the text classification problem, wherein the capsule network itself, given the feature with the symmetry hint on two ends, acts as both encoder and decoder.

1.1. Development Status of Traditional Text Classification Models

In earlier practice, text classification mainly used manually annotated data as the training set, followed by text feature extraction and feature dimensionality reduction, and finally, completed classification tasks based on feature selection results. This approach is time consuming, labor intensive, and inefficient. As early as the 1950s, authors in [9] first proposed using word frequency statistics to estimate the category of articles, that is, the frequency of text occurrence, marking the beginning of the automation stage of text classification technology. Since the 1960s, authors in [10] have published automatic text classification using keywords and proposed probability and factorization models. When using this approach, it is necessary to manually define classification rules and gain a deep understanding of the field in order to develop reasonable rules. However, the approach's method of constructing classifiers is relatively simple, its models are also simple, and its practicality is low, making it not suitable for large-scale text classification tasks. Since the end of the last century, the amount of text data on the internet has grown rapidly. Domain experts no longer need to develop classification rules based on specific domain knowledge. According to various experimental results, the text classification system based on machine learning technology is superior to traditional classification systems in terms of accuracy and speed.

The text classification method based on machine learning technology mainly involves two parts: text representation and classifiers. The classifiers include the K-nearest neighbor algorithm [11], support vector machine [12], maximum information entropy [13], etc. Authors in [14] used the LSI algorithm to calculate the similarity of one text to another. Authors in [15] classified text using the maximum likelihood estimation method and Bayesian model. Authors in [16] used vector machine algorithms to improve the performance of text classification. This paper reviewed the feature extraction, classifier design, and interrelationship between Chinese texts both in and outside of China in recent years, and it put forward some problems to be solved and future development directions.

Meanwhile, in the field of text classification, the number of corpora is also increasing. These corpora contain text data and corresponding label data in various fields, which greatly facilitates the study of text classification [17]. Owing to the rapid development of the field of artificial intelligence, machine learning as a new data analysis method has gradually been applied to text classification, making text classification more efficient and accurate [18]. At the same time, this also brings new challenges to text classification. While the machine learning text classification technique has been widely used by researchers, it has some drawbacks. For example, while it performs well when the text structure is simple and the data categories are balanced, in practice, text classification schemes tend to be more complex, such as short text in news headlines that are less informative and sometimes unevenly categorized. In these cases, the text classification model based on machine learning performs poorly.

In recent years, many new technologies have emerged in the field of text classification, achieving good results. As the field of deep learning has continued to develop, researchers have gradually turned to text classification based on this type of learning, and especially to the hybrid model convolutional neural network, which integrates the attention mechanism, recurrent neural network and convolutional neural network.

1.2. Development Status of Text Classification Models Based on CNNs

In the past decade, convolutional neural networks (CNNs) have revolutionized artificial visual perception, achieving significant image spatial dimension results in various core areas of computer vision, from image classification [19] to object detection [20] and instance segmentation [21]. Collobert and Weston [22] applied CNNs to the NLP field, starting a wave of CNN research in this field.

Authors in [23] used CNNs for text classification tasks and proposed a classic CNN text classification model involving a convolutional layer, post-connection pooling layer, and final classifier that uses a fully connected layer with dropout. At the same time, in order to pre-train word vectors on text, the Word2vec model proposed by Mikolov [24] was used, which has achieved significant results on multiple datasets.

In the same year, authors in [25] proposed a CNN model involving K-maxpooling, in which different K values are pooled at different network layers, and each layer of the network accepts the maximum feature value of the K group. Authors in [26] proposed a weakly supervised learning method based on convolutional neural networks to maximize the applicability of sentiment analysis results by distinguishing between positive and negative keywords. Authors in [27] proposed a semi-supervised CNN text classification model based on region embedding and proposed a new text embedding method called two-view embedding, which achieved good results in the sentiment classification task at the sentence level. Authors in [28] proposed a CNN text classification model that uses characters instead of words as input. Subsequently, in 2015, authors in [29] proposed a character-level input neural network language model, in which a CNN is used to encode characters for learning, and the output of the CNN is used as the input for a recurrent neural network language model. Authors in [30] proposed a CNN text classification method based on sentence-level supervised learning. For deep CNN text classification, authors in [31] referred to the image domain CNN model and proposed a deep CNN text classification model called VDCNN, which learns long-distance text information by stacking convolutional layers and expanding convolution kernels. This model has achieved good results on long text datasets.

The capsule network (CapsNet) can be regarded as a CNN variant [32]. In the CapsNet, a capsule is defined as a set of neurons with instance parameters. The length of the vector represents the probability of the presence of a feature, meanwhile, the direction of the vector represents different types of instantiated parameters. The information flow among capsules is achieved by the so-called routing algorithm. The self-attention mechanism had already been implemented into the CapsNet in [33,34] for image recognition. However, the CapsNet with self-attention mechanism had significantly fewer parameters [33]. In light of the above investigation, we were motivated to propose a newly designed framework combining CapsNet with a self-attention mechanism applied to text classification.

In summary, traditional text classification methods rely on artificial features, which have the problem of being sparse and having high dimensions. Meanwhile, in recent years, deep learning has faced certain problems when being applied to both shallow and deep CNN text classification models.

1.3. Research Content of This Article

According to our analysis of the current state of the literature both in and outside of China, it is evident that few researchers have paid attention to the efficiency of capsule networks and their ability to better represent object transformations internally [35]. In fact, all model solutions proposed to date have considered a large number of parameters, an approach which inevitably conceals the inherent generalization ability of capsules. In this paper, an SA-CapsNet model is proposed, which improves on the original CapsNet model while having fewer parameters. It was found to be able to achieve state-of-the-art results on three different datasets, preserving all important aspects of the capsule network. The proposed solution utilizes the similarity between low-level capsules to cluster and route them to more promising high-level capsules. The main contributions of this model are summarized as follows:

(1) We conducted in-depth research on the generalization ability of capsule-based networks, analyzed the advantages and problems of capsule networks in text classification, and reduced the number of trainable parameters compared to previous studies.

(2) Our conceptualization and development of capsules based on efficient and highly replicable deep learning neural networks was found to be able to achieve state-of-the-art results on three different datasets.

(3) We tested the SA CapsNet model on publicly available text datasets, and we compared the model to text classification models based on CNNs, RNNs, LSTM, etc., in order to verify its effectiveness.

2. Relevant Theoretical Foundations

2.1. Text Classification Model Based on Deep Learning

Traditional text classification methods, such as the logistic regression, support vector machine, naive Bayes, decision tree methods, etc., mostly adopt statistical thinking. Faced with abstract text data, statistical machine learning methods can only extract shallow information, making it difficult for them to capture deep semantic information. Compared with traditional text classification methods, deep-learning-based text classification methods can learn deeper and more complex semantic features. In such methods, word vectors are input into a deep learning model. Then, different deep learning models are used to extract semantic information from the text, obtain the feature vectors of the text, and finally obtain the probability of each category through a classifier. The CNN model and RNN sequence model are widely used in the field of text classification [36].

2.2. Text Classification Model Based on Convolutional Neural Networks

In 2014, Kim proposed a convolutional neural network model [37] that successfully applied convolutional neural networks to the field of text classification. The processing method for one-dimensional text data differs from that for images. Kim utilized multiple convolutional kernels of different sizes to obtain different local text representations, ultimately achieving good model performance.

The structure of the text classification model based on convolutional neural networks is shown in Figure 1. The model adopts a trained k-dimensional word vector with a single sentence of n words, and the input to the model is a matrix of dimensions $n * k$. The model's second layer is a convolutional layer, which uses convolutional kernels of different sizes and quantities to capture information in the text from different perspectives. The size of the convolutional kernel is $m * k$. Each different convolutional kernel converts the input matrix into a vector. The model's last layer is the pooling layer, which maximizes the vector output from each convolutional kernel through convolution operations and outputs it as a scalar. Then, all pooling results are concatenated to obtain a vector. By fully connecting this vector to a softmax layer with an equal number of text categories, the model obtains the classification results of the text.

From the above diagram, it can be seen that the CNN model extracts n-gram like information from the text through convolutional kernels, and then obtains more accurate sentence vector information through the pooling layer, achieving good results. However, the model does not take into account the global information of the entire document and finds it difficult to capture the order of words. These are still some of the problems that the CNN faces in text classification.

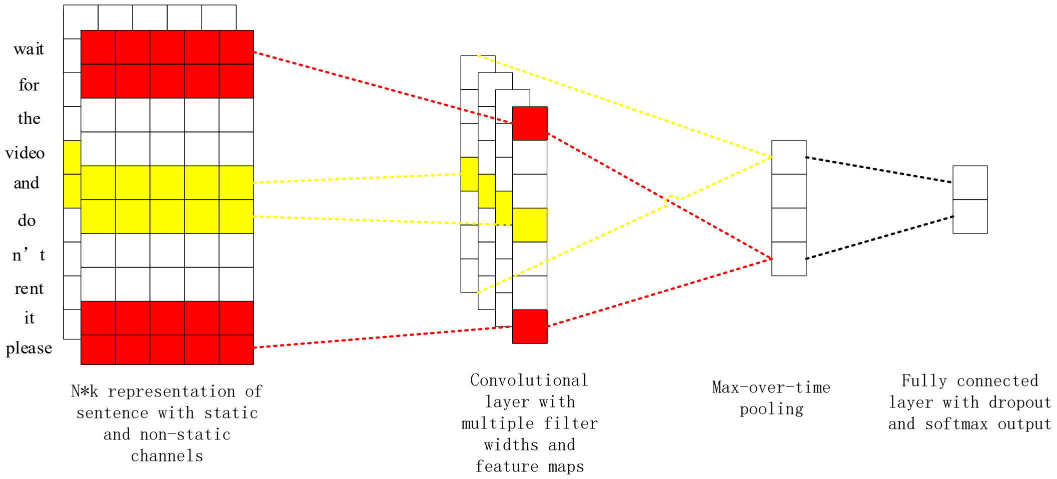
wait
for
the
video
and
do
n' t
rent
it
please
N*k representation of sentence with static and non-static channels
Convolutional layer with multiple filter widths and feature maps
Max-over-time pooling
Fully connected layer with dropout and softmax output

Figure 1. Structure diagram of convolutional neural network model for text classification.

2.3. Text Classification Model Based on Recurrent Neural Networks

In both feedforward neural networks and convolutional neural networks, their neurons are connected to the neurons in the previous layer. These networks have certain limitations in their application to non-stationary signal processing. For example, the CNN structure is relatively complex and requires a large amount of computation to learn the mapping relationships between samples; furthermore, it requires a long training time, among other issues. Sequential data such as voice, text, and video are correlated in time series. Faced with such a series of data, the performance of CNNs is not satisfactory.

Recurrent neural networks, which are neural network structures with memory functions, are one of the most fundamental models used for deep learning. They exhibit strong parallelism and parallel computing capabilities in both the temporal and spatial dimensions. Unlike traditional neural networks, recurrent neural networks use a multi-layer feedforward structure to organize the relationships between input variables in order to achieve efficient parallel operations. The neurons in the RNN layer not only receive information from the upper layer neurons, but also from the same layer neurons. Therefore, RNNs are naturally suitable for processing sequential data, such as text and speech. RNNs not only can overcome the window size limit faced by CNNs, but also have no requirements for text length, theoretically being able to handle any length of text.

The RNN model applied to text classification is shown in Figure 2. Each word in the text is mapped to a word vector, which is then input into the RNN layer for feature extraction. Finally, the output of the RNN layer is fully connected to the softmax layer for text classification.

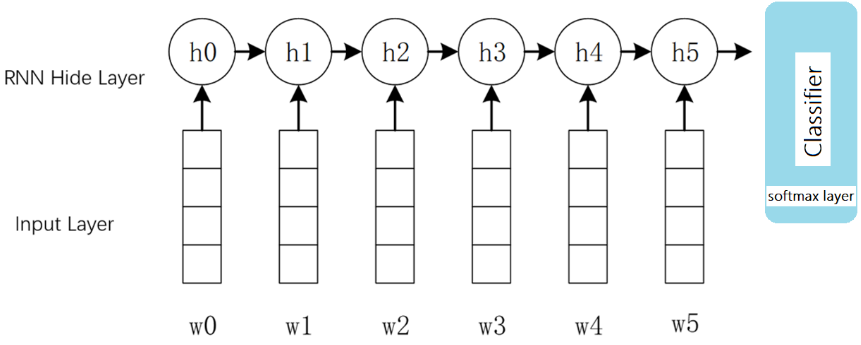

Figure 2. Structure of text classification model based on recurrent neural network.

2.4. Text Classification Model Based on Attention Mechanism

2.4.1. Principle of Attention Mechanism

Attention models (AM) are currently an important component of neural network structures in the AI field. They have been widely used in fields such as natural language processing, statistical learning, speech, and computer science.

The attention mechanism used in deep learning is closely related to the encoder–decoder framework. Under this framework, learners can not only selectively pay attention to information and extract attention, but also utilize varying degrees of injected attention to improve their learning outcomes. The encoder–decoder framework without the attention mechanism is shown in Figure 3.

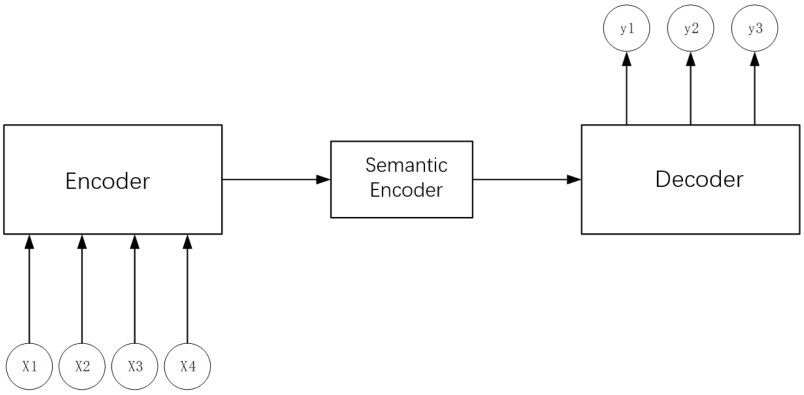

Figure 3. Encoder–decoder framework without attention mechanism.

The attention mechanism can be intuitively explained using human visual mechanisms. For example, a human's visual system is more inclined to focus on the auxiliary judgment information in an image and ignore irrelevant information. Similarly, in issues involving language or vision, certain parts of the input may be more helpful than others for decision making. In the encoder–decoder framework without the attention mechanism, the generation of each one depends on the same semantic encoding C, meaning that the model is operating similar to the human eye when it is unable to focus when looking at something. Therefore, this method is significantly different from the actual situation, making its effect not satisfactory.

The encoder–decoder framework with the attention mechanism is shown in Figure 4. In this framework, the function of the target that predicts the output word sequence becomes $y_i = G(C_i, y_1, y_2, \cdots y_{i-1})$, each C_i of which may correspond to a different attention probability distribution in the source word sequence. The G function is a nonlinear function $C_i = \sum_{j=1}^{L_x} a_{ij} h_j$ established using a neural network, where h_j is the result of encoding the encoder layer x_j, and aj is the output C_i for h_j's level of attention, that is, the h_j weighted average, which is the result of attention distribution.

In the attention mechanism, the input sequence can be viewed as a combination of many key values. When the query and key in the given output sequence are calculated, the corresponding weights are obtained, and then the weights are weighted and summed on the values in the input sequence to obtain the desired result, as Equation (1) shown in below.

$$Attention(Query, Source) = \sum_{j=1}^{L_x} S(Query, key_i) * Value_i \quad (1)$$

where L represents the length or size of a sequence being considered (in NPL), and x in L_x represents the dimensionality.

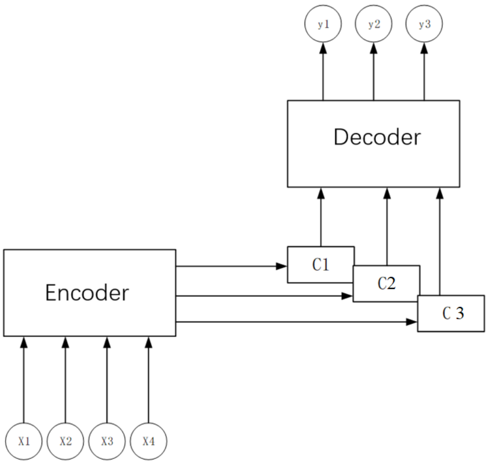

Figure 4. Encoder–decoder framework with attention mechanism.

2.4.2. Text Classification Model Based on Attention Mechanism

The several attention-mechanism-based text classification models introduced in this paper provide new ideas for solving text classification problems.

Authors in [38] proposed a hierarchical attention network for text classification. This model has two obvious characteristics: (1) it has a hierarchical structure model based on the natural hierarchical structure of the text, and (2) it establishes attention mechanisms at both the word and sentence levels, enabling it to use attention mechanisms for different levels of content when constructing text representations. The performance of the model was found to be superior to existing methods in six text classification tasks. Authors in [39] utilized the hierarchical attention mechanism to apply the hierarchical attention model to cross language emotion classification. In this mechanism, the word level attention mechanism learns which word in a sentence better represents the meaning of the sentence.

Authors in [40] proposed a directed self-attention network. This network can only learn sentence embedding based on the proposed attention mechanism. Authors in [41] turned the problem of text classification into a problem of matching labels with words. Authors in [42] proposed a sentence to match the attention mechanism of CNN models. Currently, how to automatically discover implicit information in text from a large number of datasets is a very meaningful area of research. Two effective methods are proposed to achieve this: first, directly using machine learning algorithms; second, using deep neural network technology. They studied three attention mechanisms that integrate the inter-relationships between sentences into a CNN. Through the validation of multiple text classification tasks, including answer selection and implicit text meaning tasks, they showed that these sentences are expressed better than individual sentences.

In addition, authors in [43] used a self-attention model to extract interpretable statement embeddings. However, these models all have the common drawback that they cannot effectively extract contextual semantic information. Authors in [44] proposed a CNN model that introduces attention mechanisms in order to better extract text features. Authors in [45] utilized the attention mechanism of the entity bag model to perform text classification tasks. The Levenberg–Marquardt algorithm is an effective text representation method, but it requires a predetermined threshold to determine whether to train and how to learn the optimal classifier, making it difficult for the method to achieve good performance. Authors in [46] utilized attention mechanisms to decompose complex problems into simpler sub problems. Authors in [2] explored universal pooling in order to better obtain sentence embeddings and proposed a vector-based attention mechanism model.

2.5. Classification Model Evaluation Indicators

After obtaining classification results using a text classification model, a universal standard needs to be used to evaluate the results. The most commonly used evaluation criteria include the accuracy, recall, and F1 value criteria. The confusion matrix structure is shown in Table 1.

Table 1. Confusion matrix.

Forecast Results	Actual Results	
True	TP	FP
False	FN	TN

The specific evaluation indicators of these criteria are as follows:

Accuracy—The proportion of correctly classified samples compared to the total sample size, calculated as follows:

$$Accuracy = \frac{TP + TN}{TF + FP + FN + TN} \tag{2}$$

Precision—The ratio of the number of correctly predicted and actually classified samples to the number of correctly predicted and classified samples, calculated as follows:

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

Recall rate—The proportion of the number of correctly predicted and actually classified samples compared to the number of correctly classified samples, calculated as follows:

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

The $F1$ value takes into account both accuracy and recall, and its calculation expression is as follows:

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{5}$$

3. Self-Attention Capsule Network

3.1. The Overall Framework of the Model

This paper proposes a self-attention capsule network (SA-CapsNet) text classification model with a self-attention mechanism. The overall framework of the model is shown in Figure 5.

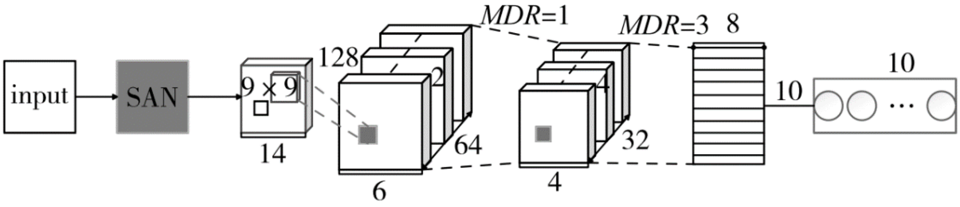

Figure 5. SA-CapsNet modeling framework.

The input of the SA-CapsNet model is a fixed length text sequence sample, and its output is a specific category. The SA-CapsNet model's framework can be divided into two parts: a self-attention module and a capsule module.

The self-attention module weights the extracted features to obtain a stronger feature representation for the capsule vector, and it runs this under the self-attention algorithm to route low-level capsules to the whole they represent.

In the capsule module, the SA-CapsNet model adds an intermediate capsule layer and reduces the dimensionality of the capsule before increasing it. Through the primary capsule layer and the intermediate capsule layer, the feature information is processed to remove noise, and the vector neurons are compressed in a direction-invariant manner using a squash function, compressing the size of the vector to within the range of 0–1.

In the middle of the capsule module, dynamic routing algorithms are used to score and predict advanced features based on posture relationships through low-level capsules, which selectively activate high-level capsules. The features extracted through the capsules can include many different types of instantiation parameters, as well as the presence of the features themselves.

In this study, the length of each vector was used to represent the probability represented by the entity represented by the capsule. This approach is compatible with self-attention routing algorithms and does not require any reasonable objective function minimization.

3.2. Self-Attention Mechanism

The self-attention mechanism is an important part of the proposed SA-CapsNet model. This mechanism focuses on the scaled dot product attention mechanism. The scaled dot product attention mechanism is used to calculate the degree of association between the current word and other words, and based on this degree of association, to obtain the weight vector of the current word and other words, and finally to use this weight vector as a feature representation.

3.2.1. Multi-Head Attention Mechanism

The multi-head attention mechanism structure, which is equivalent to the operation of multiple scaling point multiplication attention mechanisms, is used to obtain various feature representations, making the extracted features of the model more diverse and thereby improving the model's feature extraction ability. Figure 6 shows the structure of the multi-head attention mechanism. The mechanism maps Q (Query), K (Key), and V (Value) in a linear manner, and then scales the h data in parallel to double the attention mechanism. The results are then checked and linearly transformed to obtain the final result.

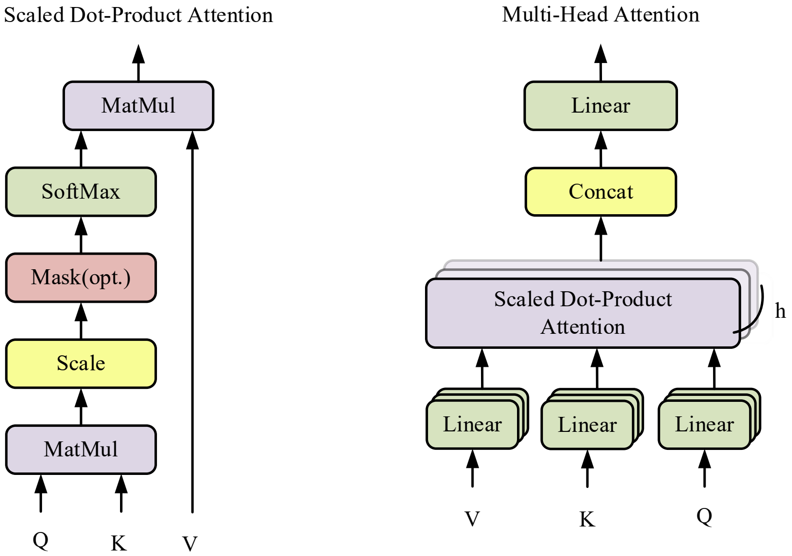

Figure 6. Self-attention structure diagram.

The structure of the scaling dot multiplication attention mechanism is shown in Figure 7. Each word in the text is represented by three vectors: Q, K, and V. Multiplying the input vector X with different initialized weight matrices yields Q, K, and V, as shown in Equations (6) to (8).

$$Q = X \otimes W^Q \tag{6}$$

$$K = X \otimes W^K \tag{7}$$

$$V = X \otimes W^V \tag{8}$$

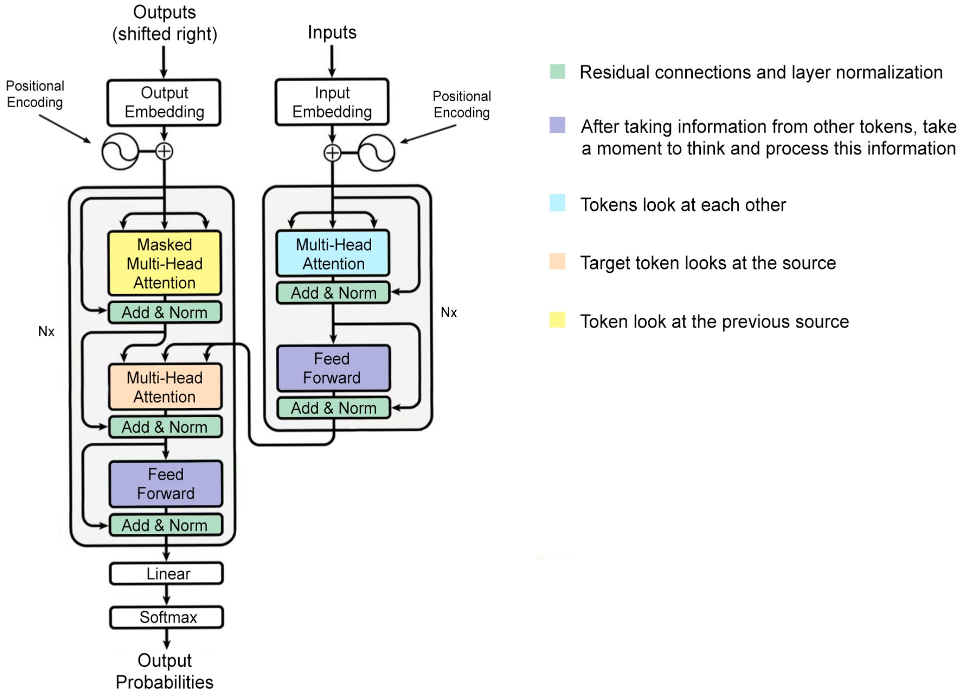
Outputs
(shifted right)
Inputs
Positional Encoding
Output Embedding
Input Embedding
Positional Encoding
Masked Multi-Head Attention
Add & Norm
Multi-Head Attention
Add & Norm
Nx
Multi-Head Attention
Add & Norm
Feed Forward
Add & Norm
Nx
Feed Forward
Add & Norm
Linear
Softmax
Output Probabilities
Residual connections and layer normalization
After taking information from other tokens, take a moment to think and process this information
Tokens look at each other
Target token looks at the source
Token look at the previous source

Figure 7. Overall structure of transformer.

Then, based on the Q, K, and V vectors, the results of the scaling point multiplication attention mechanism are calculated, as shown in Equation (9).

$$Attention(Q, K, V) = softmax\left(\frac{Q \times K^T}{\sqrt{d_k}}\right)V \tag{9}$$

where the d_k is an extra dimension K for adjustment that prevents the inner product from becoming too large in the similarity calculation.

Essentially, the zoom point multiplication attention mechanism involves calculating the relationship between the current word and all remaining words in the text, thereby obtaining the mutual contribution to the current word, weighting all words in the text, and then performing a weighted summation. Due to this attention mechanism acting on all the words contained in the text itself, the mechanism is also known as the self-attention mechanism. By utilizing the zoom magnification attention mechanism, a multi-head attention mechanism can be constructed on this basis.

3.2.2. Transformer Module

The proposed capsule network text classification model with the self-attention mechanism was introduced in Section 3.1. In this section, the transformer module is described in detail. The innovation in the transformer module is that it only utilizes the self-attention mechanism and does not require sequential execution to obtain high-quality text features. Therefore, the module can perform parallel computations and has higher computational efficiency. In addition, it adds a position embedding vector to provide text position information for the model. The experimental results of the present study show that this method effectively improves the model's computational efficiency. In addition, the Transformer module also has good fault tolerance and can resist varying degrees of noise interference.

The structure of the transformer is shown in Figure 7. The encoder consists of six identical encoding blocks. In addition, the module also uses convolutional neural networks as feature extraction tools and combines contextual information for semantic analysis. The decoder also includes six decoding blocks. Each encoder block includes the following structures: the self-attention mechanism, feedforward network, residual connection, and normalization mechanism. The self-attention mechanism also includes the multi-head attention and scaled dot product attention structures.

3.2.3. Location Encoding

With regards to the self-attention mechanism, it can be seen that it has no structure that captures sequence information, which means that even if the order of a sentence is disrupted, the features captured by the transformer are the same. This is unacceptable for some natural processing tasks. In order to avoid this situation and reflect the positional relationships among words in the text, positional vectors are added to the transformer's word vector layer, as shown in Figure 8.

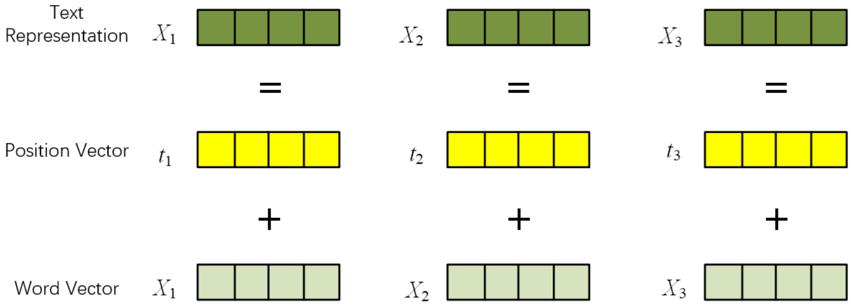

Figure 8. Text representation in transformer.

The self-attention mechanism is a complete information extractor that eliminates the use of the traditional RNN and other information extraction units, thus effectively avoiding some of the inherent defects of RNN units and making the entire model more concise. For example, in the process of attention computation, the self-attention mechanism uses matrix multiplication to correlate the original information between two pairs, which avoids the long-distance information loss caused by gradient vanishing in recurrent neural networks such as the GRU network and avoids the problem of RNN units being difficult to parallelize.

The transformer module, which utilizes the self-attention mechanism as its core idea, has also been validated for its powerful performance through various experiments. From the experimental results, it can be seen that the transformer's ability to extract semantic features, long-distance features, and comprehensive features is stronger than that of the CNN and RNN.

From Table 2, when the sequence length and feature dimension are the same order of magnitude, the time complexity of the self-attention mechanism and the RNN was better than that of the CNN. In the table, n is the sequence length, d is the feature dimension, and k is the kernel size of convolutions.

Table 2. Comparison of self-attention mechanism, RNN, and CNN computing efficiency.

Model	Computing Complexity in One Layer
Self-Attention	$\mathrm{O}\left(n^2 * d\right)$
RNN	$\mathrm{O}\left(n * d^2\right)$
CNN	$\mathrm{O}\left(k * n * d^2\right)$

3.3. Capsule Module

An important feature of the proposed SA-CapsNet model is its ability to extract more expressive features using capsule vectors. Capsule vectors are used to solve the problems of low coding efficiency and the loss of position information between features during pooling operations in convolutional neural networks. Capsule networks differ from convolutional neural networks in that they use vector capsules instead of the neurons that are aggregated by convolutional neural networks, squash functions instead of ReLU activation functions, and dynamic routing instead of pooling operations. A capsule module diagram is shown in Figure 9.

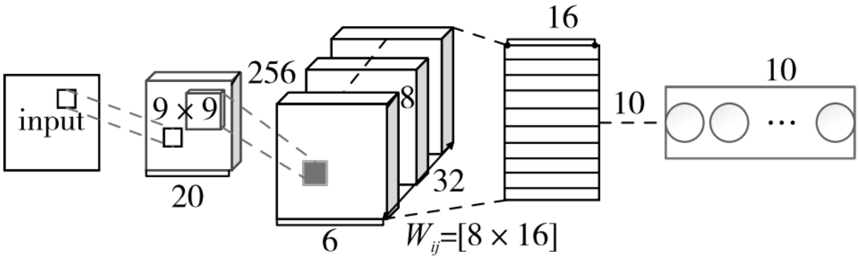

Figure 9. Capsule module, which is used to extract text features with spatial layer relationship information.

The core structure of the capsule module is composed of three layers. The first two layers are convolutional operation layers; text features are obtained through these two layers. Then, by reconstructing the text features of adjacent units into capsule vector representations, the text features with spatial hierarchical relationship information are obtained. Further to this, by constructing a digital capsule layer, a vector with a dimension of 16 is set to represent specific categories. Among these categories, the length of the vector in the digital capsule layer can represent the probability of entity existence.

3.3.1. ReLU Activation Function

In the case of a single instance, before entering the main capsule layer, a set of convolutional and batch normalization layers are extracted by extracting local features. Each output of the convolutional layer is activated by convolutional operations with a specific kernel dimension k, feature mapping number f, step size s = 1, and ReLU as the activation function, as follows:

$$F^{l+1}\left(X^{l}\right) = ReLU\left(Conv_{k*k}\left(X^{l}\right)\right) \tag{10}$$

Overall, the first convolutional part of the network can be modeled as a single function H_{conv}. This function maps the input to a higher dimensional space, promoting the creation of capsules. On the other hand, a part of the second network is the primary tool used by primary capsules to create vector representations of the features they represent. This is a deeply separable convolution with linear activation, and it only performs the first step of the deep space convolution operation, acting on each channel separately. In addition, imposing the kernel dimension $k * k$ and the quantity equal to the output dimension H × W filters f and H_{conv}, following which the function F can obtain the primary capsule layer $S^{l}_{n,d}$, where n^{l} and d^{l} are the numbers for the primary capsules in the first layer and their respective sizes.

3.3.2. Dynamic Routing Mechanism

Capsule networks cluster input features by using dynamic routing instead of the pooling layer operations used in convolutional neural networks. The more similar the features are, the stronger these features become. Therefore, in the proposed model, feature selection is carried out to achieve the purpose of pooling layer feature selection. The steps of the dynamic routing algorithm (as shown in Algorithm 1) are as detailed below.

The length of the output vector of a capsule is to represent the probability that the entity represented by the capsule is present in the current input. A non-linear function is used to normalize the short vectors to almost zero length and long vectors to a length as follows:

$$\boldsymbol{v}_j = \frac{\left\|\boldsymbol{s}_j\right\|^2}{1+\left\|\boldsymbol{s}_j\right\|^2}\frac{\boldsymbol{s}_j}{\left\|\boldsymbol{s}_j\right\|} \tag{11}$$

where $\boldsymbol{v}_j$ is the output vector of capsule j and $\boldsymbol{s}_j$ is its total input. For all the first layers of capsules, the total input to a capsule $\boldsymbol{s}_j$ is a weighted sum over all prediction vectors

$\hat{u}_{j|i}$ from the capsules in the layer below and is produced by multiplying the output $\boldsymbol{u}_i$ of a capsule in the layer below by a weight matrix $\boldsymbol{W}_{ij}$.

$$s_j = \sum_i c_{ij}\hat{\boldsymbol{u}}_{j|i} \tag{12}$$

$$\hat{\boldsymbol{u}}_{j|i} = \boldsymbol{W}_{ij}\boldsymbol{u}_i \tag{13}$$

where the c_{ij} are coupling coefficients that are determined by the iterative dynamic routing process. The coupling coefficients between capsule i and all the capsules in the layer above sum to 1 and are determined by a "**routing softmax**" function as follows:

$$c_{ij} = \frac{\exp\left(b_{ij}\right)}{\sum_k \exp\left(b_{ij}\right)} \tag{14}$$

$$\textstyle\sum_j c_{ij} = 1 \tag{15}$$

where initial logits b_{ij} are the log prior probabilities that capsule i should be coupled to capsule j.

Algorithm 1 Capsule Network Dynamic Routing Algorithm

```
Input $\hat{\mathbf{u}}_{j|i}, r, l$
for all capsule I in layer l and capsule j in layer (l + 1): $b_{ij} = 0$
t = 0
while t < r do
    for all capsule i in layer l: $\mathbf{c}_i = routingSoftmax(\mathbf{b}_i)$
    for all capsule j in layer (l + 1): $\mathbf{s}_j = \sum_i c_{ij}\hat{\mathbf{u}}_{j|i}$
    for all capsule j in layer (l + 1): $\mathbf{v}_j = squash(\mathbf{s}_j)$
    for all capsule i in layer l and capsule j in layer (l + 1): $b_{ij} = b_{ij} + \hat{\mathbf{u}}_{j|i} \cdot \mathbf{v}_j)$
$t = t + 1$
end while
Output $\mathbf{v}_j$
```

3.4. Loss Function

The output layer is represented by a vector. In fact, not only does the capsule in the last layer represent the existence of a specific probability object class, but all its attributes are also extracted from its various parts. The vector is used to represent the probability of the existence of capsule entities. Its length should be close to 1 only when the entity it represents uniquely exists. Therefore, in order to allow for multiple classifications, Formula (16) is calculated for each class represented by the capsule n^L of the last layer L:

$$L_{n^L} = T_{n^L}\max\left(0, m^+ - \left\|u_n^L\right\|\right)^2 + \lambda\left(1 - T_{n^L}\right)\max\left(0, \left\|u_n^L\right\| - m^-\right)^2 \tag{16}$$

If there is a class n^L and m^+, m^-, and if there are hyperparameters to be adjusted, then T_{n^L}, are equal to 1. Then, the individual marginal losses L_{n^L} are added together to calculate the final score for the training stage. Finally, a reconstruction regularizer is used to ensure that the final capsule has robustness and meaningful attributes.

4. Experimental Design and Analysis

The goal of this paper was to simply demonstrate that a functioning capsule network should reduce the number of parameters required in a text classification model by better embedding information due to its inherent ability. The proposed method was tested in an experimental environment to evaluate its accuracy compared to traditional convolutional neural networks and similar methods. To this end, three commonly used text datasets were used to test the proposed method based on network evaluation: the IMDB, MPQA, and MR datasets. All the experiments showed that capsule networks with self-attention mechanisms can achieve better results than other comparable methods.

4.1. Experimental Dataset and Preprocessing

This study conducted experiments on three benchmark datasets, which specific details and parameter statistics are shown in Table 3. In the table, Dataset refers to the dataset name; Train corresponds to the number of training set samples; Valid refers to the number of validation set samples; Test corresponds to the number of samples in the test set; Class refers to the number of target categories; Arg.T represents the average length of the sample sentences; Max.T represents the maximum length of the sample sentence; and Vocabsize represents the size of the vocabulary.

Table 3. Benchmark dataset table.

Dataset	Class	Train	Valid	Test	Arg.T	Max.T	Vocabsize
IMDB	2	20,000	5000	25,000	238	2494	10,000
MPQA	2	6362	2121	2121	3	34	2661
MR	2	6398	2132	2132	16	53	16,540

(1) IMDB Available online: https://www.imdb.com/interfaces/ (accessed on 5 April 2024) dataset [47]: This was an IMDB English film review dataset integrated internally by Keras. The experiment used a labeled dataset containing 50,000 IMDB film reviews with significant bias, specifically for emotional analysis. The 25,000 reviews marked with a training set did not include movies in the 25,000 reviews test set. Among these movies, 25,000 were used as training sets and 25,000 were used as testing sets, and the labels used were pos (positive) and neg (negative). In addition, the positively and negatively labelled reviews were equal in number.

(2) MPQA Available online: https://mpqa.cs.pitt.edu/ (accessed on 5 April 2024) dataset [48]: This was a binary dataset that mostly consisted of various English news articles. Among these articles, there were a total of 3311 positive tendencies and 7293 negative tendencies. During the experiment, 80% of the data were used to form a training set and the remaining 20% were used as a testing set.

(3) MR Available online: http://www.cs.cornell.edu/people/pabo/movie-review-data/ (accessed on 5 April 2024) dataset [49]: This was a binary dataset that was mostly sourced from short texts from professional English film review websites containing the emotional tendencies of “positive” and “negative”, with the dataset containing 5331 entries each in these categories. In this study, the dataset segmentation method used was the same as that used in the MPQA dataset.

Comparing these three different datasets, the sample sentences in the IMDB dataset were relatively long; the average length is 238 words [47]. While those in the MPQA and MR datasets were relatively short; the average length is 20 words [48,49].

The specific experimental details and parameter statistics are shown in Table 4. As can be seen, the Activation parameter corresponds to the extrusion function using ReLU; Optimizer corresponds to the optimizer Adam; Loss corresponds to the loss function Margin; Input_ Size corresponds to a batch size of 200; n layers correspond to four layers of the stack; Total params corresponds to the total parameter quantity of 3,452,864; EPS corresponds to a neural network learning rate of 10^{-7}; and Epoch represents the number of training rounds.

After determining the dataset, it is necessary to preprocess the text data when transmitting it to the network. As the data in this study were segmented comments, it was necessary to first segment the text data. After tokenization is performed, the words need to be converted into numbers in order to form a vector matrix that is transmitted to the network one number at a time. In this study, words were converted into numbers using the subscript of the word in the dictionary as the number representing the word. Through this process, the text data were transformed into vector data, and then the vector data were all normalized to the same length. We chose the maximum length of the vectors, while vectors less than the specified length were filled with 0. In addition, this study treated all punctuation marks, special characters, and other language symbols that appeared in the dataset as spaces and changed all uppercase letters in the dataset to lowercase letters. Further to this, during the process of building the vocabulary, words that only appeared once in the dataset were deleted.

Table 4. Experimental parameters.

Parameters	Setting
Activation	ReLU
Optimizer	Adam
Loss	Margin
Input_size	200
n layers	4
Total params	3,452,864
Eps	1×10^{-7}
Epoch	50

4.2. Experimental Results

This study analyzed the performance of the proposed SA-CapsNet model with regards to its accuracy of classification results. First, the classification accuracy of the model under different datasets was calculated, as shown in Table 5.

Table 5. Test performance of different datasets.

Dataset	TP Number	FP Number	Accuracy/%
IMDB	21,180	3820	84.72
MPQA	8519	2085	80.31
MR	8036	2624	75.38

From the above table, it can be seen that the SA-CapsNet model performed well on the different datasets, recording an accuracy of over 75%, while it achieved an accuracy of over 80% on both the IMDB and MPQA datasets. Taking the IMDB dataset as an example, the model's accuracy and margin loss are shown in Figures 10 and 11.

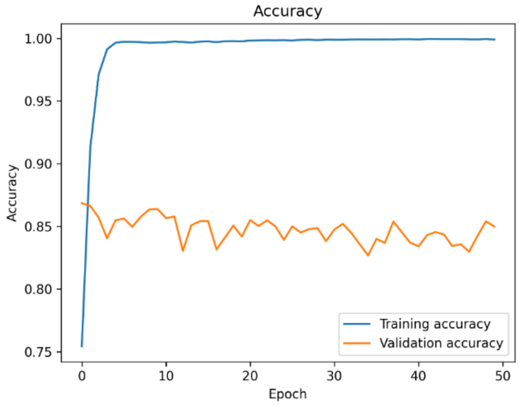

Figure 10. Accuracy.

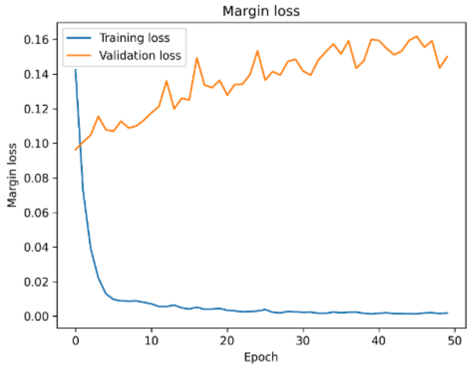

Figure 11. Margin loss.

4.3. Ablation Experiment

Next, to verify the advantages of the SA-CapsNet model, the study conducted relevant experiments on the model's self-attention network and capsule layer structures, as well as on the fusion of the two. These experiments were conducted by mapping self-attention networks with different structures to feature extraction layers of capsule networks, and the experimental results are shown in Table 6.

Table 6. Comparative experiments on self-attention networks with different structures.

Model Name	IMDB/%	MPQA/%	MR/%
Capsule Network	83.64	79.92	73.95
1SA + 2Maxpooling + CapsNet	83.95	80.01	74.52
2SA + 2Maxpooling + CapsNet	84.34	80.51	75.20
3SA + 2Maxpooling + CapsNet	84.72	80.31	75.38
2SA + 2Maxpooling + CapsNet	84.42	79.98	75.81

As can be seen from the above results, when using the self-attention module (SA) as the basic unit in constructing a self-attention network, increasing the number of SA layers achieved a better fit and model data training results, up until three SA layers were used. When three layers were used, overfitting gradually occurred as the training parameters increased, leading to a decrease in classification accuracy. At present, pooling operations can cause extracted features to lose some spatial information. Indeed, when the pooling operations were reduced once in these experiments, the classification accuracy was improved. However, without pooling operations, a model's accuracy will decrease due to there being redundant information in the features. Therefore, a self-attention network structure was ultimately used to add one layer of pooling to a three-layer SA. To verify the effectiveness of this proposed model, ablation experiments were conducted, as shown in Table 7.

Table 7. Network structure ablation experiment.

Num	Model Name	IMDB/%	MPQA/%	MR/%
1	CNN + CapsNet	83.72	79.68	73.27
2	SAN + CapsNet	83.92	79.96	74.55
3	CNN + NCapsNet	84.35	80.05	75.27
4	SA + NCapsNet	84.72	80.31	75.38

Comparing the results of experiments 1, 2, 3, and 4, it can be seen that mapping the self-attention module to the feature extraction layer of the capsule network improved the classification accuracy of the model. Additionally, it can be seen that increasing the middle capsule layer and reducing the dimensionality of the capsule not only reduced the number of parameters but also improved the performance of the model.

4.4. Comparative Experiment

Next, a comparison experiment was performed. In this experiment, three classic networks, namely, the CNN, RNN, and LSTM network, were tested on the three datasets. The classification accuracy results for these three networks and the capsule network used in this study are shown in Table 8.

Table 8. The comparison of accuracy.

Model Name	IMDB	MPQA	MR
CNN	78.86	73.52	70.36
RNN	80.98	74.98	71.36
LSTM	82.28	76.36	72.68
Capsule Network	83.64	79.92	73.95
SA-CapsNet	84.72	80.31	75.38

Firstly, a CNN is mainly composed of convolutional layers and pooling layers. Because convolutional operations can effectively extract image feature information, CNNs are widely used by researchers. In most cases, researchers use them on image datasets, and they are commonly employed in image classification, image segmentation, and other tasks; they can also be used for text classification. In this study, a CNN was used on text datasets. The performance accuracy of this model on the three datasets was 78.86%, 73.52%, and 70.36%, respectively, meaning its performance was relatively poor. This is because CNNs have a small field of view problem, with their perception actually being the size of the convolutional kernel, which is often set to five or three. Although their perception can be increased by stacking convolutional layers, CNNs may also encounter other problems, such as gradient disappearance or gradient explosion, so the use of CNNs is therefore not suitable for text datasets.

RNNs are specifically designed to process data with temporal characteristics. The output of an RNN's neurons is not only transmitted to the next layer, but also retains a hidden state, which is passed on to the next neuron until it is changed in subsequent divine elements. This hidden state gives subsequent neurons the opportunity to see the state preserved by previous neurons far away from them; this is very important in text classification as such classification involves a section of a sentence being analyzed. Usually, the subject of the sentence appears at the beginning, and the meaning or verb conveyed by the sentence often appears in the middle. The implicit state of an RNN is conducive to the network identifying the sentences in the later part while connecting these with the subject appearing at the beginning in order to make judgments. In this study, the performance accuracy of the RNN on the four datasets was 80.98%, 74.98%, and 71.36%, respectively. Therefore, compared to the CNN, the RNN achieved a higher accuracy, owing to its unique hidden state.

The LSTM network is a recurrent neural network based on the short-term memory model. It is based on RNNs and has characteristics that match its name, being designed to solve long-distance problems. In addition to the hidden characteristic possessed by RNNs, the LSTM network adds three additional gate attributes, termed the forgetting gate, memory gate, and output gate. The forgetting gate is used to calculate the information to be forgotten, the memory gate is used to calculate the information to be memorized, and the output gate combines the previously obtained information to calculate the output information. The three gate attributes complement each other, enabling the LSTM network to remember words that are far away from the current position. At the same time, the presence of the forgetting gate allows the network to ignore some unimportant information. The performance accuracy of the LSTM network on the three datasets was 82.28%, 76.36%, and 72.68%, respectively. Therefore, compared to the RNN, its text classification accuracy was better, thanks to its ability to remember long and short distance feature information.

As a comparison with the original experiment, CapsNet was also tested on the three datasets. CapsNet, also known as the capsule network, was developed in order to solve the problems faced by CNNs. It was first applied to image classification and has gradually achieved good performance results in some classification task experiments. The capsule network is applied to text classification models and proposed two structures: the first uses single-scale features in the convolutional layer, and the second uses multi-scale features in the convolutional layer. Their experiments showed that multi-scale features are superior to single-scale features, as these contain richer and more diverse grammatical information. However, this finding ignores the fact that the various scale features corresponding to words within a text should not be equally important. In this study, the performance accuracy of CapsNet on the three datasets was 83.64%, 79.92%, and 74.95%, respectively.

Finally, SA-CapsNet had a performance accuracy of 84.72%, 80.31%, and 75.38%, respectively, on the three datasets. Therefore, compared to the other three models, SA-CapsNet's accuracy was superior by more than 2%; this was thanks to its unique self-attention mechanism capsule attribute.

5. Conclusions

5.1. Summary

Text classification is the most fundamental task in natural language processing. Due to the rapid development of artificial intelligence, the amount of text information on the internet has exploded in recent times. Against the backdrop of unprecedented success regarding the deep learning method, there has been a recent research surge in the field of natural language processing. Furthermore, in recent years, the transformer structure has been widely used in this field and has been proven to be a better network structure than that of the recurrent neural network. Therefore, owing to this structure's text representation and classification performance, it is necessary to develop network structures based on the transformer structure.

This study focused on reducing the parameters of the model for various applications, such as edge computation and mobile applications. This study proposed a capsule network with a self-attention mechanism for text classification based on CapsNet. The proposed network integrates a self-attention mechanism (SA) with good global information extraction ability, enabling the model to achieve better feature extraction ability by allowing the capsule to carry more feature information. Further to this, the model simplifies CapsNet and improves its noise filtering ability by reconstructing the capsule network, reducing the capsule dimension, and adding an intermediate capsule layer.

To test the performance of the proposed SA-CapsNet model, this study trained and tested it on three datasets: the IMDB, MPQA, and MR datasets. Compared to the original CapsNet model, the proposed model's classification accuracy on the three datasets was still improved, namely by 1.08%, 0.39%, and 1.43%, respectively. The experimental results show that the accuracy is still maintained with the reduced parameters.

5.2. Future Research Prospects

This study conducted in-depth research on the attention mechanism and text classification. Based on the deep learning architecture, and through combining the self-attention mechanism with the capsule network, an SA-CapsNet model was developed, which achieved good results on experiments featuring English datasets. However, the following issues still need further exploration:

(1) Further research on text representation and the classification of multi-label tasks and long text data is required. Due to the experimental conditions and time constraints, this study only conducted experiments using binary classification tasks and small sample datasets, and the experimental data balance was poor, resulting in some of its conclusions having certain limitations. In order to comprehensively and objectively verify the performance of the proposed model, it should be experimentally verified on a larger number of complex text classification and long text datasets.

(2) Research based on Chinese text representation and text classification is necessary. Due to the fact that the datasets used to validate the proposed model in this study

were all English text datasets, the model's performance is yet to be tested on other languages (such as Chinese). It is therefore necessary to conduct further experiments using Chinese and other datasets.

(3) In-depth text classification research involving deep learning that features, for instance, the RCNN model and other hybrid models that integrate the recurrent neural network and convolutional neural network is still required.

Author Contributions: Conceptualization, X.Y., S.-N.L. and T.-W.K.; methodology, S.-N.L.; software, Z.C.; validation, Y.W.; formal analysis, X.Y. and T.-W.K.; investigation, S.-N.L.; resources, X.Y. and T.-W.K.; data curation, S.-N.L.; writing—original draft preparation, X.Y.; writing—review and editing, T.-W.K. and S.-P.T.; visualization, Z.C.; supervision, X.Y.; project administration, S.-N.L.; funding acquisition, T.-W.K., S.-N.L. and X.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research and the APC were partially funded by Sanda University Grant number [D021202.22.058], [2021ZD05] and [2022BC088].

Data Availability Statement: Data are contained within the article.

Acknowledgments: This research is supported by the school research funding, Sanda University. We are grateful to Shaozhong Zhu, Chen Yiyi, Pan Hui-bin and Chen Ying for their great supports on this project to fund the new methods in the study of neural networks.

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

References

1. Ashish, V.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems 30, Long Beach, CA, USA, 4–9 December 2017.
2. Chen, Q.; Ling, Z.H.; Zhu, X. Enhancing sentence embedding with generalized pooling. *arXiv* **2018**, arXiv:1806.09828.
3. Galassi, A.; Lippi, M.; Torroni, P. Attention in natural language processing. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *32*, 4291–4308. [CrossRef] [PubMed]
4. Chen, C.W.; Tseng, S.P.; Kuan, T.W.; Wang, J.F. Outpatient text classification using attention-based bidirectional LSTM for robot-assisted servicing in hospital. *Information* **2020**, *11*, 106. [CrossRef]
5. Rosen, J. Symmetry at the Foundation of Science and Nature. *Symmetry* **2009**, *1*, 3–9. [CrossRef]
6. Haykin, S. Neural networks expand SP's horizons. *IEEE Signal Process. Mag.* **1996**, *13*, 24–49. [CrossRef]
7. Alfonso Perez, G.; Colchero Paetz, J.V. Velocity Estimations in Blood Microflows via Machine Learning Symmetries. *Symmetry* **2024**, *16*, 428. [CrossRef]
8. Shavlokhova, V.; Vollmer, A.; Stoll, C.; Vollmer, M.; Lang, G.M.; Saravi, B. Assessing the Role of Facial Symmetry and Asymmetry between Partners in Predicting Relationship Duration: A Pilot Deep Learning Analysis of Celebrity Couples. *Symmetry* **2024**, *16*, 176. [CrossRef]
9. Edmundson, H.P.; Wyllys, R.E. Automatic abstracting and indexing—Survey and recommendations. *Commun. ACM* **1961**, *4*, 226–234. [CrossRef]
10. Maron, M.E.; Kuhns, J.L. On relevance, probabilistic indexing and information retrieval. *J. ACM* **1960**, *7*, 216–244. [CrossRef]
11. Peterson, L.E. K-nearest neighbor. *Scholarpedia* **2009**, *4*, 1883. [CrossRef]
12. Tong, S.; Koller, D. Support vector machine active learning with applications to text classification. *J. Mach. Learn. Res.* **2001**, *2*, 45–66. [CrossRef]
13. Li, R.; Tao, X.; Tang, L.; Hu, Y. Using maximum entropy model for Chinese text categorization. In Proceedings of the Advanced Web Technologies and Applications: 6th Asia-Pacific Web Conference, APWeb 2004, Hangzhou, China, 14–17 April 2004; Proceedings 6; Springer: Berlin/Heidelberg, Germany, 2004; pp. 578–587.
14. Zelikovitz, S.; Marquez, F. Transductive learning for short-text classification problems using latent semantic indexing. *Int. J. Pattern Recognit. Artif. Intell.* **2005**, *19*, 143–163. [CrossRef]
15. Wawre, S.V.; Deshmukh, S.N. Sentiment classification using machine learning techniques. *Int. J. Sci. Res. (IJSR)* **2016**, *5*, 819–821.
16. Thelwall, M.; Buckley, K.; Paltoglou, G. Sentiment in Twitter events. *J. Am. Soc. Inf. Sci. Technol.* **2011**, *62*, 406–418. [CrossRef]
17. Luo, W. Research and implementation of text topic classification based on text CNN. In Proceedings of the 3rd International Conference on Computer Vision, Image and Deep Learning & International Conference on Computer Engineering and Applications (CVIDL & ICCEA), Changchun, China, 20–22 May 2022; pp. 1152–1155.
18. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient estimation of word representations in vector space. *arXiv* **2013**, arXiv:1301.3781.
19. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]

20. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
21. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
22. Collobert, R.; Weston, J.; Bottou, L.; Karlen, M.; Kavukcuoglu, K.; Kuksa, P. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.* **2011**, *12*, 2493–2537.
23. Chen, Y. Convolutional Neural Network for Sentence Classification. Master's Thesis, University of Waterloo, Waterloo, ON, Canada, 2015.
24. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed representations of words and phrases and their compositionality. *Adv. Neural Inf. Process. Syst.* **2013**, *26*, 1–9.
25. Kalchbrenner, N.; Grefenstette, E.; Blunsom, P. A convolutional neural network for modelling sentences. *arXiv* **2014**, arXiv:1404.2188.
26. 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]
27. Johnson, R.; Zhang, T. Semi-supervised convolutional neural networks for text categorization via region embedding. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 919–927. [PubMed]
28. Zhang, X.; Zhao, J.; LeCun, Y. Character-level convolutional networks for text classification. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 2–9.
29. Kim, Y.; Jernite, Y.; Sontag, D.; Rush, A. Character-aware neural language models. In Proceedings of the AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016.
30. Zhai, Z.L.; Zhang, X.; Fang, F.F.; Yao, L.Y. Text classification of Chinese news based on multi-scale CNN and LSTM hybrid model. *Multimed. Tools Appl.* **2023**, *82*, 20975–20988. [CrossRef]
31. Conneau, A.; Schwenk, H.; Barrault, L.; Lecun, Y. Very deep convolutional networks for text classification. *arXiv* **2016**, arXiv:1606.01781.
32. Parikh, A.P.; Täckström, O.; Das, D.; Uszkoreit, J. A decomposable attention model for natural language inference. *arXiv* **2016**, arXiv:1606.01933.
33. Li, H. Deep learning for natural language processing: Advantages and challenges. *Natl. Sci. Rev.* **2018**, *5*, 24–26. [CrossRef]
34. Maas, A.; 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, Portland, OR, USA, 19–24 June 2011; pp. 142–150.
35. Pang, B.; Lee, L. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. *arXiv* **2005**, arXiv:cs/0506075.
36. Chen, C.W.; Chung, W.C.; Wang, J.F.; Tseng, S.P. Application of Multiple BERT Model in Construction Litigation. In Proceedings of the 2020 8th International Conference on Orange Technology (ICOT) IEEE, Daegu, Republic of Korea, 18–21 December 2020; pp. 1–4.
37. Mikolov, T.; Karafiát, M.; Burget, L.; Cernocký, J.; Khudanpur, S. Recurrent neural network-based language model. In Proceedings of the Interspeech, Chiba, Japan, 26–30 September 2010; pp. 1045–1048.
38. 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.
39. Zhou, X.; Wan, X.; Xiao, J. Attention-based LSTM network for cross-lingual sentiment classification. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, TX, USA, 1–5 November 2016; pp. 247–256.
40. Shen, T.; Zhou, T.; Long, G.; Jiang, J.; Pan, S.; Zhang, C. Disan: Directional self-attention network for rnn/cnn-free language understanding. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018.
41. Wang, G.; Li, C.; Wang, W.; Zhang, Y.; Shen, D.; Zhang, X.; Henao, R.; Carin, L. Joint embedding of words and labels for text classification. *arXiv* **2018**, arXiv:1805.04174.
42. Yin, W.; Schütze, H.; Xiang, B.; Zhou, B. Abcnn: Attention-based convolutional neural network for modeling sentence pairs. *Trans. Assoc. Comput. Linguist.* **2016**, *4*, 259–272. [CrossRef]
43. Lin, Z.; Feng, M.; Santos CN, D.; Yu, M.; Xiang, B.; Zhou, B.; Bengio, Y. A structured self-attentive sentence embedding. *arXiv* **2017**, arXiv:1703.03130.
44. Wang, S.; Huang, M.; Deng, Z. Densely connected CNN with multi-scale feature attention for text classification. In Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), Stockholm, Sweden, 13–19 July 2018; pp. 4468–4474.
45. Yamada, I.; Shindo, H. Neural attentive bag-of-entities model for text classification. *arXiv* **2019**, arXiv:1909.01259.
46. Deng, L.; Wiebe, J. Mpqa 3.0: An entity/event-level sentiment corpus. In Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, CO, USA, 31 May–5 June 2015; pp. 1323–1328.
47. Sabour, S.; Frosst, N.; Hinton, G.E. Dynamic routing between capsules. In Proceedings of the Advances in Neural Information Processing Systems 30, Long Beach, CA, USA, 4–9 December 2017.

48. Shang, Y.; Xu, N.; Jin, Z.; Yao, X. Capsule network based on self-attention mechanism. In Proceedings of the 2021 13th International Conference on Wireless Communications and Signal Processing (WCSP) IEEE, Changsha, China, 20–22 October 2021; pp. 1–4.
49. Mazzia, V.; Salvetti, F.; Chiaberge, M. Efficient-capsnet: Capsule network with self-attention routing. *Sci. Rep.* **2021**, *11*, 14634. [CrossRef] [PubMed]

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