Text3D: 3D Convolutional Neural Networks for Text Classification

Jinrui Wang, Jie Li * and Yirui Zhang

Academic Editor: Yung-Lyul Lee

Received: 21 June 2023
Revised: 13 July 2023
Accepted: 14 July 2023
Published: 16 July 2023

Article

Abstract: Convolutional Neural Networks (CNNs) have demonstrated promising performance in many NLP tasks owing to their excellent local feature-extraction capability. Many previous works have made word-level 2D CNNs deeper to capture global representations of text. Three-dimensional CNNs perform excellently in CV tasks through spatiotemporal feature learning, though they are little utilized in text classification task. This paper proposes a simple, yet effective, approach for hierarchy feature learning using 3D CNN in text classification tasks, named Text3D. Text3D efficiently extracts rich information through text representations structured in three dimensions produced by pretrained language model BERT. Specifically, our Text3D utilizes word order, word embedding and hierarchy information of BERT encoder layers as features of three dimensions. The proposed model with 12 layers outperforms the baselines on four benchmark datasets for sentiment classification and topic categorization. Text3D with a different hierarchy of output from BERT layers demonstrates that the linguistic features from different layers have varied effects on text classification.

Keywords: text classification; 3D convolution; pretrained language model

1. Introduction

Text classification is a fundamental problem in Natural Language Processing (NLP). Many studies can be defined as text classification tasks, such as topic categorization, sentiment analysis and relation extraction. Deep neural networks have been widely used in text classification tasks since word order information can be utilized and more semantic features can be captured, compared with traditional bag-of-words/n-grams models [1].

A CNN is a feedforward network consisting of convolution layers and pooling layers. While simple and shallow convolutional neural networks (CNNs) [2,3] were proposed earlier, deeper and more complex neural networks have also been studied. Examples are deep character-level CNNs [4,5], a complex combination of CNNs and recurrent neural networks (RNNs) [6] and RNNs in a word-sentence hierarchy [7]. The Convolutional Neural Network family has some influential architectures. Essentially, the convolution layer converts every small patch of data into vectors at every location, which can be in parallel. The data can be either the original text/image or the output of the previous layer and the location can be a three word window around each word. There have been several recent studies of CNN for text categorization in large data settings. For example, Conneau et al. [5] have found that very deep 32-layer character-level CNNs outperform deep 9-layer character-level CNNs. However, very shallow 1-layer word-level CNNs [3] were shown to be more accurate and much faster than the very deep 32-layer character-level CNNs. Zhang et al. [8] extended Kim Yoon’s [2] model and empirically identified the hyper parameter setting. Kim’s TextCNN [2] inspired part of the method proposed in this paper. DPCNN [9] uses the deepening of word level neural network to capture the global representation of the text, and proposes a deep pyramid convolutional neural network, which achieves excellent accuracy by increasing the network depth. A special Convolutional Neural Network, ResNet [10], proposed by Microsoft Research, performs well in image-classification and object-detection tasks. The characteristic of ResNet is that...
it is easy to optimize and can improve the accuracy of result by considerably increasing depth. The internal residual blocks of ResNet use skip connections, which alleviates the vanishing gradient problem caused by increasing the depth in neural network. VGG [11] is a work proposed by the Visual Geometry Group team at Oxford, focusing on the impact of convolutional network depth on the accuracy of large-scale image recognition. The highlight of the network is the use of multiple small convolutional filters (3 × 3) replacing the large-scale convolution kernel to reduce the number of layers of the network. The original motivation of the Inception [12] method proposed by Google may be ‘want all’. We either use many filters of the same size or add a pooling layer in a regular CNN, while the Inception method aggregates all these operations in single layer.

Going further than the 2D CNNs mentioned above, 3D CNNs have the ability for extra feature learning. Previous work [13] has confirmed that 3D CNNs are good feature-learning machines in the field of CV because of spatiotemporal feature learning. However, 3D CNN is little utilized in text classification task because the feature of the third dimension is not easy to decide.

For classification tasks, the linguistic features provided to the classifier are crucial as they encode relevant information about the input in the embedding space. Recently, large pretraining language models such as ELMO [14], BERT [15], XLNet [16] and so on have also shown their outstanding performance in all kinds of NLP tasks, including text classification. Transformer-based architectures [17] and BERT [15] in particular achieve remarkable results in natural language processing. The experiments have shown that BERT outperforms previous state-of-the-art models in eleven NLP tasks in GLUE benchmark [18] by a significant margin. For classification tasks, the most common approach is to feed the [CLS] representation obtained from the last layer (the hidden state of the [CLS] token from the last transformer encoder layer) to softmax [19], which outputs the probability of a label. In addition to BERT, many researchers also focus on other pretrained language models. Pre-training or fine-tuning BERT on downstream datasets may be computationally expensive because it contains a large number of parameters and the effectiveness behind it is the ambiguity. ALBERT [20] proposed a lightweight version of BERT that uses parameter reduction technology to reduce memory consumption, which shows that there is a significant improvement in the scaling of multi statement input. On the other hand, RoBERTa [21] presented a study on BERT pretraining, and applied alternative pretraining objectives and tasks, showing their (and pretraining data) impact on performance. XLNet [16] improved BERT using auto-regressive pretraining, and achieved bidirectional contextual learning using expected likelihood, and addressed the limitations of BERT caused by ignoring the dependencies between masked positions. SMART [22] proposed a learning framework that uses smooth induced regularization and Bregman proximal point optimization to achieve effective fine-tuning and better generalization. SMART+RoBERTa [22] and SMART+BERT [22] show performance improvements over BERT [15] and RoBERTa [21] in various NLP tasks.

However, The BERT-based methods and techniques proposed in the past few years in the natural language processing field have not been combined in any way with 3D Convolutional Neural Networks which are widely used in the Computer Vision field. Word embedding learned by the last layer of BERT is directly fed to a simple linear classifier in previous text classification tasks, ignoring structural information about language learned by BERT. Jawahar et al. [23] have found that layers of BERT encode a rich hierarchy of linguistic information on inputting words, with surface features in bottom layers, syntactic features in middle layers and semantic features in top layers.

We propose an effective approach for hierarchy feature learning in text classification, using three-dimensional convolution network (Text3D). Text3D has a multi-filters structure that focuses on contexts between words and connections between hierarchies. The novel Text3D utilizes word order, word embedding and hierarchy information of BERT encoder layers as features of three dimensions. We empirically show that these learned features
with a simple linear classifier can yield good performance on text classification tasks. The contributions of this paper are summarized as follows:

- We propose a three-dimensional convolution network (Text3D) which uses word order, word embedding and hierarchy information of BERT encoder layers as three dimension features.
- We utilize a well-designed 3D convolution mechanism and multiple filters to capture text representations structured in three dimensions produced by pretrained language model BERT.
- We conduct extensive experiments on datasets of different scales and types. The results show that our proposed model produces significant improvements on baseline models. Furthermore, we conduct additional experiments to confirm that hierarchy features of different BERT layers encode different types of word knowledge.

2. Our Approach

**Overview of Text3D:** We try to identify a good architecture for 3D ConvNets. Our Text3D classification model is composed of two parts: text representation and a 3D CNN model. The overall structure can be seen in Figure 1. We fix the word-receptive field to \( h \times k \times s \), where \( s \) is the depth of the 3D convolution kernels. Text representation is produced by multiple layers of BERT encoder. The key constructions of Text3D are as follows:

- Text representation within three dimensions, including word order, word embedding and hierarchy information of BERT.
- Well-designed 3D convolution mechanism and filters to extract features of three dimensions from text representation.

**Figure 1.** The structure of Text3D classification model, which is composed of text representation and a 3D CNN model.

2.1. Text Representation

The word embedding is constructed by an input encoder (such as BERT). As shown in Figure 1, the text to be classified (a sequence of input words) is inputted to the input encoder to generate the input text representation. The encoding process is shown in the first part of Figure 1 and Equation (1).

\[
x_1, x_2, \cdots, x_n = \text{Encoder}(w_1, w_2, \cdots, w_n)
\]  

(1)

The output of different encoder layers contains different hierarchies of word knowledge. The text representation of three dimensions will be fed to the Text3D to predict the text category.

2.2. Model Architecture

The model architecture of our Text3D is shown in the second part of Figure 1. BERT’s embedding is used in our approach as the word vector. Let \( x_i \in \mathbb{R}^k \) be the \( k \)-dimensional word vector corresponding to the \( i \)-th word in a sentence of length \( n \). The sentence is represented as:

\[
x_{i:n} = x_1 \oplus x_2 \cdots \oplus x_n
\]  

(2)
where $\oplus$ is an operator conducting concatenation. A convolution progress involves a kernel $v \in \mathbb{R}^{hks}$, which is applied to a cube filter of $h$ words, BERT representation of $k$ dimensions for each word and $s$ BERT layers to produce a new feature. A 2D convolution applied to an image will output an image. Meanwhile, a 2D convolution applied to multiple images also results in an image as multiple images are treated as different channels [24]. Hence, 2D ConvNets only capture two dimension features of the input representation in every convolution operation. Compared to 2D ConvNet, 3D ConvNet has the ability to model richer information owing to 3D convolution and 3D pooling operations. Our Text3D preserves the hierarchy information of the input signals produced by BERT, which will result in an output volume (three dimension features). A feature $c_i$ is generated through a cube kernel from words $x_{i:i+h-1}$ by:

$$c_i = f(W \cdot x_{i:i+h−1} + b)$$  \hspace{1cm} (3)

Here $b \in \mathbb{R}$ is a bias term and $f$ is a non-linear function such as the hyperbolic tangent. This kernel is applied to each possible words in the sentence $\{x_{1:h}, x_{2:h+1} \cdots, x_{n−h+1:n}\}$ to produce a feature map:

$$c = [c_1, c_2 \cdots, c_{n−h+1}]$$ \hspace{1cm} (4)

with $c \in \mathbb{R}^{(n−h+1)(d−s+1)}$. We then apply a max-overtime pooling operation [25] over the feature map and take the maximum value $\hat{c} = max(c)$ as the feature corresponding to this particular kernel. The idea is to capture the most important feature with the highest value for each feature map.

The model uses multiple filters (with varying cube sizes) to obtain multiple features. These features form the penultimate layer and are fed to a fully connected softmax layer whose output is the probability distribution over labels. In the single channel architecture, illustrated in Figure 1, filters are applied and the results are added to calculate $c_i$ in Equation (3). We employ dropout on the second last layer with a constraint on l2-norms of the weight vectors [26] for regularization. Dropout prevents co-adaptation of hidden units by randomly dropping out—i.e., setting to zero—a proportion $p$ of the hidden units during forward–back propagation.

3. Experiments

We evaluate our proposed Text3D on two different text classification tasks, including sentiment analysis and topic categorization. We report the experiments with Text3D in comparison with previous models and baseline. We will make the code publicly available.

3.1. Datasets and Baselines

We use four widely-studied datasets compiled by Zhang et al. [4] to evaluate our model. These datasets are diverse in the aspects of type, size, number of classes and document length. Table 1 shows the statistics of the datasets. For sentiment analysis, we use Yelp Review Polarity (Yelp P.) which is the reviews subset of Yelp Open Dataset. Yelp P. is a binary sentiment classification dataset whose class is either positive or negative. For topic categorization, we use AG’s News (AGNews), DBPedia and Yahoo! Answers (Yahoo). AG’s News contains news articles from the AG’s corpus of news articles on the web pertaining to the four largest classes. DBPedia is an ontology classification dataset containing 14 non-overlapping categories picked from DBpedia 2014. Yahoo consists of questions and answers from the “Yahoo! Answers” website. Baseline models are as follows:

- **TextCNN** [2]: a popular CNN-based classifier exploiting one-dimensional convolution operation and max overtime pooling.
- **CNN-based models** [4]: word-level CNN and char-level CNN.
- **BERT** [15]: the pretrained BERT-based-uncased model (fine-tuned) with classifier.
Table 1. The four text classification datasets used in our experiments.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Type</th>
<th>Classes</th>
<th>Train</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>AGNews</td>
<td>News Topic</td>
<td>4</td>
<td>120 K</td>
<td>7.6 K</td>
</tr>
<tr>
<td>DBpedia</td>
<td>Wikipedia Topic</td>
<td>14</td>
<td>560 K</td>
<td>70 K</td>
</tr>
<tr>
<td>Yelp Polarity</td>
<td>Sentiment</td>
<td>2</td>
<td>560 K</td>
<td>38 K</td>
</tr>
<tr>
<td>Yahoo</td>
<td>QA Topic</td>
<td>10</td>
<td>1.4 M</td>
<td>60 K</td>
</tr>
</tbody>
</table>

3.2. Experiment Settings

The hyper-parameters involved are set as follows. The CNN-based models have a filter size of [3, 4, 5] with 100 filters of each. Word embedding matrix for CNN-based models is initialized with 300-d Glove word embedding [27]. The employed BERT model is the BERT-base-uncased, including 12 layers, 768 hidden units and 110 M parameters. We adopt Adam optimizer. We use GeLUs [28] for all the nonlinear activation functions. For our approach, Dropout regularization [29] with a rate of 0.25 is applied during training. The learning rate for center loss is $2 \times 10^{-5}$. Batchsize is set to 32 and the model is trained for 5 epochs. The 12 BERT layers are all adopted as the third dimension in the 3D convolution networks. In the results below, the depth of the filter is fixed to 2 and the stride is set to 1 unless otherwise specified. As for the computation configuration for the experiments, we adopted NVIDIA A30.

3.3. Test Performance

We first report the accuracy of our full model (Text3D with 12 BERT layers) on the datasets (Table 2). To put it into perspective, we also show the previous results in the table. In general, our proposed model consistently improves the performance of the baseline models (i.e., TextCNN, CNNs and BERT) on all datasets. Furthermore, Text3D outperforms all of the previous results on all the topic datasets and ranks the 2nd best on sentiment dataset, which validates the effectiveness of our approach.

Table 2. Accuracy on all datasets. The best results are shown in bold.

<table>
<thead>
<tr>
<th>Model</th>
<th>AGNews</th>
<th>DBpedia</th>
<th>Yelp Polarity</th>
<th>Yahoo</th>
</tr>
</thead>
<tbody>
<tr>
<td>TextCNN</td>
<td>0.8808</td>
<td>0.9610</td>
<td>0.9453</td>
<td>0.4966</td>
</tr>
<tr>
<td>Word-level CNN</td>
<td>0.9145</td>
<td>0.9858</td>
<td>0.9540</td>
<td>0.7116</td>
</tr>
<tr>
<td>Char-level CNN</td>
<td>0.9015</td>
<td>0.9834</td>
<td>0.9475</td>
<td>0.7120</td>
</tr>
<tr>
<td>BERT</td>
<td>0.9329</td>
<td>0.9806</td>
<td>0.9453</td>
<td>0.7635</td>
</tr>
<tr>
<td>FastText [30]</td>
<td>0.9250</td>
<td>0.9860</td>
<td>0.9570</td>
<td>0.7230</td>
</tr>
<tr>
<td>DPCNN [9]</td>
<td>0.9313</td>
<td>0.9912</td>
<td>0.9736</td>
<td>0.7610</td>
</tr>
<tr>
<td>Text3D</td>
<td>0.9380</td>
<td>0.9919</td>
<td>0.9580</td>
<td>0.7695</td>
</tr>
</tbody>
</table>

Experiment results on the four text classification datasets are shown in Table 2, which is the classification accuracy on sentiment analysis and topic categorization tasks. Our proposed Text3D achieves the best performance among all the baseline models. For Yelp P. dataset, Text3D ranks the 2nd best and has comparable performance with DPCNN [9], a deep pyramid convolution neural networks for text categorization.

Compared with CNN-based models, Text3D performs better than TextCNN, shallow CNN (word/character-level) and deep CNN (DPCNN) with a significant margin in topic categorization tasks. The reason is that Text3D is able to capture structural features, which is the extra dimension not contained in 2D CNN. Comparing to the deep 2D convolution neural network, utilizing an extra dimension of features is more efficient in small datasets than only increasing the layers of neural network. The shallow convolution network has greater advantages in model architecture, computation complexity and computation efficiency.

Compared with the fine-tuned BERT model, Text3D greatly outperforms the fine-tuning BERT in four datasets and two text classification tasks (sentiment analysis and topic categorization). Text3D respectively exceeded the baseline BERT in accuracy by 0.51%, 1.13%, 1.17% and 0.60%. The 3D convolution neural network makes full use of representations from different layers of BERT owing to the feature-extraction ability. The
feature-extraction ability results in Text3D outperforming the fine-tuned BERT in text classification task. Structural information generated by different BERT layers can be integrated through the filters in 3D CNN. The 3D convolution neural network makes more use of BERT embedding. Feature extraction mechanism using both token order and semantic representation of BERT performs better than a single classifier following behind BERT.

Both TextCNN and other CNN-based models (Char-level CNN et al.) of baselines are influential works on text classification using 2D convolutional network. The test performance indicates that the improvements to the baseline models are significant. For the small dataset AGNews, our method outperforms TextCNN by 5.71%, exceeding DPCNN by a margin of 0.66%. When it comes to larger dataset Yahoo, our method achieves the best performance over the baselines, outperforming the CNNs by 5.75%. The Text3D increases the accuracy by 3.09% on DBpedia compared with TextCNN. Text3D produces substantial improvements in the accuracy of pretrained BERT model, i.e., 1.27% on Yelp P., which verifies the effectiveness of the proposed framework.

4. Discussion

In this section, we present more in-depth analyses and discussions on Text3D. We study the influence of some important hyper-parameters on the performance of Text3D, including different hierarchies of BERT layers and depth of kernel. We analyze the effect of hierarchy information from BERT in an ablation study.

4.1. Different Hierarchy of BERT

We conduct research to test model performance with different hierarchies of BERT layers. We take bottom (1–4), middle (5–8) and top (9–12) hierarchies of BERT layers respectively to be the word-knowledge feature in our approach. Our proposed 3D CNN method, with a depth of 4, uses output from different hierarchies of BERT (bottom, middle and top) as the third dimension of the 3D convolutional network. While the normal Text3D with the depth of 12 uses all 12 layers output information of BERT. Hyper-parameters in experiment settings are the same as in the test experiments above and the depth of kernel size is set to 2. Experiment results are shown in Table 3 and Figure 2. In Figure 2, the horizontal axis coordinates are different datasets, and the vertical axis coordinates are the accuracies of the results.

It is observed that the hierarchy of BERT layers affects the performance significantly. The accuracy results demonstrate that using output from bottom hierarchy of BERT layers as the third dimensional in 3D CNN leads to the lowest accuracy result on four datasets. Moreover, 3D CNN using top hierarchy of BERT layers outperforms the same architecture using other hierarchies in most datasets. For most datasets, the top hierarchy of BERT layers achieves the best performance. For larger QA topic dataset Yahoo, convolution with all 12 BERT layers has the optimal performance. This result in Yahoo may indicate that, for datasets with larger volume, the larger number of parameters in network model does more good for the feature extraction. Higher BERT layers are able to handle the long-distance dependency problems caused by the longer sequence of words intervening between the subject and the verb, better than the lower layers [23]. This highlights the contribution of higher layers of BERT on performing complex NLP tasks. The results confirm the conjecture that different word linguistic features are distributed over different layers. The top hierarchy encodes semantic information which contains rich word knowledge.

Table 3. Accuracy of Text3D with different hierarchy of BERT.

<table>
<thead>
<tr>
<th></th>
<th>AGNews</th>
<th>DBpedia</th>
<th>Yelp Polarity</th>
<th>Yahoo</th>
</tr>
</thead>
<tbody>
<tr>
<td>1–4</td>
<td>0.9309</td>
<td>0.9902</td>
<td>0.9132</td>
<td>0.7585</td>
</tr>
<tr>
<td>5–8</td>
<td>0.9568</td>
<td>0.9918</td>
<td>0.9524</td>
<td>0.7641</td>
</tr>
<tr>
<td>9–12</td>
<td>0.9410</td>
<td>0.9920</td>
<td>0.9599</td>
<td>0.7687</td>
</tr>
<tr>
<td>All 12</td>
<td>0.9380</td>
<td>0.9919</td>
<td>0.9580</td>
<td>0.7695</td>
</tr>
</tbody>
</table>
4.2. Depth of Filter

The depth of filter (convolution kernel) is a controlling hyper-parameter to decide the importance of convolution on the original model. Table 4 and Figure 3 show accuracy of Text3D (all 12 BERT layers) with filter depth of 2, 4, 6 and multi-depth [3, 4, 5]. In Figure 3, the horizontal axis coordinates represent different filter depths, and the vertical axis coordinates are the accuracies of the results. The four sub-graphs in Figure 3 respectively illustrate the results on AGNews, DBpedia, Yelp P and Yahoo datasets in sequence. Besides the filters of the same depth, we also set different depths for multiple filters in one convolution. It is observed that the depth of convolution filter affects the performance significantly.

![Figure 2. Accuracy (%) of the text classification using different hierarchy of BERT.](image)

### Table 4. Result of our model with different depths of filters.

<table>
<thead>
<tr>
<th></th>
<th>AGNews</th>
<th>DBpedia</th>
<th>Yelp Polarity</th>
<th>Yahoo</th>
</tr>
</thead>
<tbody>
<tr>
<td>Depth = 2</td>
<td>0.9379</td>
<td>0.9919</td>
<td>0.9580</td>
<td>0.7695</td>
</tr>
<tr>
<td>Depth = 4</td>
<td>0.9372</td>
<td>0.9918</td>
<td>0.9570</td>
<td>0.7692</td>
</tr>
<tr>
<td>Depth = 6</td>
<td>0.9371</td>
<td>0.9920</td>
<td>0.9587</td>
<td>0.7699</td>
</tr>
<tr>
<td>Depth = [3, 4, 5]</td>
<td>0.9375</td>
<td>0.9912</td>
<td>0.9581</td>
<td>0.7673</td>
</tr>
</tbody>
</table>

For small dataset AGNews, the optimal depth is 2. For large datasets like Yahoo, depth of 6 achieves the best performance. The larger filter depth will integrate more layers which means more word linguistic information contained in the feature map, thus may improve the influence of word knowledge. We can see from the Figure 3, for smaller dataset
AGNews, filters with smaller depth have better performance. Further increasing the feature information in extraction will increase model complexity and cause performance drop due to overfitting. However, when it comes to larger dataset Yahoo, filters with larger depth perform better. In this situation, we can apply the filters with larger depth to learn more word linguistic information. The results show different depths for multiple filters in one convolution has little improvement for the task effect. That means multiple filters varying in more than one dimensions (depth and height) have little advantage on classification effect due to overfitting caused by model complexity.

4.3. Ablation Study

We perform an ablation study to investigate the contributions of hierarchy signals of BERT. Results are shown in Table 5 and Figure 4. In Figure 4, the horizontal axis coordinates are different datasets, while the vertical axis coordinates represent the accuracy of the results. The subgraphs in Figure 4 are three datasets (AGNews, DBpedia and Yelp P) and their corresponding accuracy results, which is an enlarged version of the original image aiming at displaying them clearly. We use the BERT-base-uncased model [15], which is pretrained. The BERT model is used once for each dataset and generate [CLS] representations from all 12 layers, for all samples in our datasets. This means that if a dataset has a number of samples \( N \), this is results in dataset \( D \in \mathbb{R}^{N \times 12 \times 768} \). The embedding for each sentence \( E \in \mathbb{R}^{12 \times 768} \), where the second dimension 12, represents 12 [CLS] representations from 12 layers. We use the 12 [CLS] for 2D TextCNN text classification tasks to explore the effect of hierarchy information of BERT.

![Figure 4. Accuracy (%) of the text classification using Glove and BERT separately.](image)

We replace the BERT embedding with normal word embedding Glove where each token is unchangeable in the dictionary. The accuracy performance drops greatly in all datasets. For larger dataset Yahoo with more sentences, the unchangeable token embedding performs poorly. The specific embedding for certain word is limited and learns little knowledge. This demonstrates the advantage of utilizing the semantic embedding for text. The linguistic signals encoded by BERT actually contain richer word knowledge which can improve the effectiveness of text classification.

<table>
<thead>
<tr>
<th></th>
<th>AGNews</th>
<th>DBpedia</th>
<th>Yelp Polarity</th>
<th>Yahoo</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text3D</td>
<td>0.9380</td>
<td>0.9919</td>
<td>0.9580</td>
<td>0.7695</td>
</tr>
<tr>
<td>12[CLS]+TextCNN</td>
<td>0.9366</td>
<td>0.9914</td>
<td>0.9554</td>
<td>0.7664</td>
</tr>
<tr>
<td>Glove+TextCNN</td>
<td>0.9210</td>
<td>0.9610</td>
<td>0.9453</td>
<td>0.4966</td>
</tr>
</tbody>
</table>

Table 5. Ablation study on the contributions of BERT and 3D convolution network architecture. Comparison between normal embedding and BERT token.
Comparing the experimental results of Text3D and 12 [CLS]+TextCNN, it is not difficult to find that Text3D has significant improvements in four datasets and two types of text classification tasks. This comparing result demonstrates that 3D convolutional networks outperform 2D convolutional networks in feature extraction. Text3D not only focuses on the [CLS] information from the last layer of BERT, but also on the hierarchical information of the 12 encoder layers of BERT, which makes Text3D perform better than 2D convolutional networks that only utilize the output of the last layer of BERT.

5. Conclusions and Future Work

We introduce a three-dimensional convolution network (Text3D) for efficient text classification. By taking the advantages of the feature-capturing ability of 3D CNN, Text3D is able to fully utilize hierarchy signals produced by BERT ranging from top to bottom. Furthermore, a 3D convolution mechanism is proposed to capture different hierarchies of word knowledge in convolutional filter space. Detailed analyses show that Text3D is an efficient universal text classifier that achieves consistently good results over different baseline models, outperforming the 2D convolution networks.

This paper studies the application of the algorithm based on the pretrained language model in the traditional task of Natural Language Processing, text classification. Although our proposed Text3D is dedicated for text classification tasks where feature-extraction capability is particularly important, we will explore the potential applications of Text3D on other NLP tasks such as machine translation, text summarization and language modeling. Furthermore, the recently popular and cutting-edge Large Language Model (LLM), represented by ChatGPT, will be our future research topic. In the next stage, we will focus on using large language models to solve more complex and refined text tasks, such as sentiment analysis in text classification.

Author Contributions: J.L. contributed to the conception of the study; J.W. performed the experiment, contributed significantly to analysis and wrote the manuscript; Y.Z. helped perform the analysis with constructive discussions. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Natural Science Foundation of China grant number U22B2019.

Data Availability Statement: The data that support the findings of this study are openly available in public repository.

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

References

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(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.