IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 22, 2019, accepted April 12, 2019, date of publication April 18, 2019, date of current version May 2, 2019.

*Digital Object Identifier 10.1109/ACCESS.2019.2911850*

# Few-Shot Transfer Learning for Text Classification With Lightweight Word Embedding Based Models

**CHONGYU PAN, JIAN HUANG, JIANXING GONG, AND XINGSHENG YUAN**
College of Intelligence Science and Technology, National University of Defense Technology, Changsha 410073, China
Corresponding author: Jian Huang (nudtjHuang@hotmail.com)

This work was supported by the Chinese National Natural Science Foundation under Grant 61703417.

**ABSTRACT** Many deep learning architectures have been employed to model the semantic compositionality for text sequences, requiring a huge amount of supervised data for parameters training, making it unfeasible in situations where numerous annotated samples are not available or even do not exist. Different from data-hungry deep models, lightweight word embedding-based models could represent text sequences in a plug-and-play way due to their parameter-free property. In this paper, a modified hierarchical pooling strategy over pre-trained word embeddings is proposed for text classification in a few-shot transfer learning way. The model leverages and transfers knowledge obtained from some source domains to recognize and classify the unseen text sequences with just a handful of support examples in the target problem domain. The extensive experiments on five datasets including both English and Chinese text demonstrate that the simple word embedding-based models (SWEMs) with parameter-free pooling operations are able to abstract and represent the semantic text. The proposed modified hierarchical pooling method exhibits significant classification performance in the few-shot transfer learning tasks compared with other alternative methods.

**INDEX TERMS** Few-shot learning, transfer learning, text classification, word embedding based models, pooling strategy.

## I. INTRODUCTION

Deep learning methods have achieved great success across several domains and tasks in the past few years. However, these supervised learning models pose great demands for large amounts of labeled data in the task domain to iteratively train their thousands of attached parameters, as shown in Figure 1 (a). This severely limits their scalability to new classes due to annotation cost and even situations where numerous data examples do not exist. So motivated, there has been a recent interest in few-shot learning as depicted in Figure 1 (b), where models tend to learn knowledge from part of supervised data in the target domain and to recognize other novel categories with the support of few labeled examples. From the perspective of transfer learning [1] as shown in Figure 1 (c), both deep learning and few-shot learning methods train and test models in the same target domain due to the same data feature space. The difference between them is that deep learning methods share the same label space between train and test samples while few-shot learning disjoint. In contrast, humans are very good at leveraging knowledge across domains and recognizing objects with little supervision. For example, children are able to recognize a zebra in zoo by generalizing the concept from a single natural picture and even cartoon stick figure in a book. Inspired by the few-shot and transfer learning ability of humans, a new learning mechanism leveraging the knowledge from common source domains to special target domain with few data supporting should be investigated in future research. As shown in Figure 1 (d), few-shot transfer learning aims to recognize concepts from few labeled examples in target domain based on some common knowledge obtained from source domain.

The research on text classification, the baseline task in natural language processing, has sprung up since the creation of the word embedding that represents each word as a semantically dense vector [2], [3]. Leveraging the word embeddings, many deep architectures have been proposed to model the compositionality in text sequence, ranging from simple hidden layer operation like addition [4], [5], to more sophisticated hierarchical structure like Recurrent Neural Network (RNN) [6], [7] and Convolutional Neural Network

The associate editor coordinating the review of this manuscript and approving it for publication was Yunjie Yang.

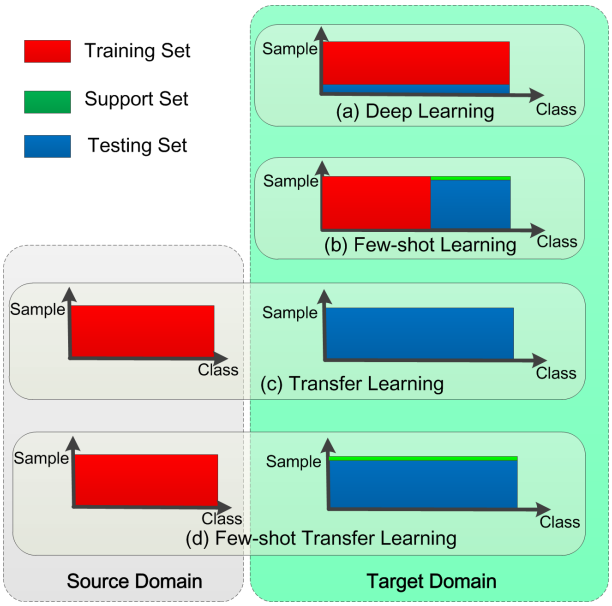



**FIGURE 1.** Training paradigm for multiple machine learning techniques. (a) As a supervised learning technique, deep learning relies on huge amounts of annotated data. (b) Trained on a part of annotated categories, few-shot learning aims to recognize new concepts with the support of few labeled samples. (c) Given a source domain and a target domain, transfer learning aims to help improve the learning of the target predictive function using the knowledge in source domain. In general, training and testing sets will share almost the same label space in transfer learning. (d) Integrating the few-shot and transfer learning ability, few-shot transfer learning tends to leverage knowledge learned from source domain to new unseen concepts in target domain with only several support examples.

(CNN) [8]–[10]. Deep models like RNNs or CNNs are typically data hungry and computationally expensive due to their thousands of parameters in expressive compositional functions, making it unfeasible in situations where numerous annotated examples are not available or even do not exist. In contrast, models with simple compositional functions could compute sentence or document representations by simply adding or averaging over the pre-trained word embeddings in a feed-forward way, making it parameter-free and efficient for implementation [4], [5]. In addition, some recent researches [11] have enlightened that simple word embedding based models could present comparative and even better performance with much simpler model structure and less parameters.

Motivated by the prevalent representation with pre-trained word embeddings and the robust generalization ability of SWEMs, we tend to explore few-shot transfer learning methods for text classification with lightweight word embedding based models. The key novelty in this paper is the combination of high-dimensional vector representations of words and the contributions are summarized in two folds:

- With only several support examples, the lightweight SWEMs are able to extract discriminative representations for text classification, both for English and Chinese documents.
- A modified hierarchical pooling method is proposed for few-shot text classification and performs best on long text datasets.

## II. RELATED WORKS

### A. FEW-SHOT TRANSFER LEARNING

There are many promising works that achieve state-of-the-art performance in visual few-shot classification domain. The Model Agnostic Meta-Learner (MAML) [12] model aims to meta-learn an initial condition for subsequent fine-tuning on few-shot problems. An optimization approach [13] goes further in meta-learning with an LSTM-based optimizer that is trained to be specifically effective for fine-tuning. Another category of metric-learning based approaches such as siamese networks [14], matching networks [15], prototypical networks [16] and relation networks [17] aim to learn a set of projection functions such that when represented in this embedding, images are easy to recognize using simple linear classifiers. Only a few works focus on the few-shot learning on NLP tasks, for example, a text classification framework based on siamese CNN network and few-shot learning is proposed in [18].

However, taking the widely used experimental setting in few-shot learning for instance, *mini*Imagenet, the benchmark dataset is split into 64, 16, and 20 classes for training, validation and testing, respectively. That is to say, in order to acquire the few-shot learning ability, all annotated examples of 64 classes in target domain are necessary while it is often unavailable in some practical applications. In contrast, few-shot transfer learning tends to remove this obstacle for simple plug-and-play applications.

### B. COMPOSITIONAL MODELS FOR TEXT CLASSIFICATION

The fundamental goal in text classification is to construct discriminative and computationally efficient compositional representation that can capture the linguistic structure of natural language sequences. Different from stacking more recurrent or convolution layers in deep models, some simple word embedding based architectures exhibit even superior performance with attention mechanism [19], [20]. More related works are Deep Averaging Network (DAN) [4] or fastText [5], where averaging over word embeddings achieves promising results on some NLP tasks. Generally, such a simple word embedding based model does not explicitly account for spatial, word-order information within a text sequence. However, they possess the desirable property of having significantly fewer parameters, enjoying much faster training, relative to RNN or CNN based models. A recent method for few-shot text classification was proposed in [21] where documents are represented using a simple weighted average of constituent word embeddings. It is based on the sentence embedding method Smooth Inverse Frequency (SIF) proposed in [22]. Furthermore, a recent work [11] has proposed a hierarchical pooling operation to incorporate spatial information and further investigated when and why simple

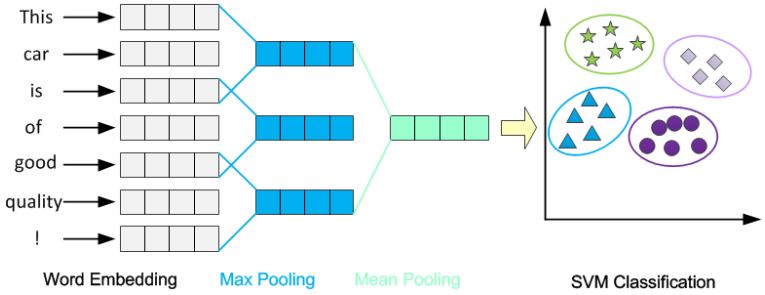



**FIGURE 2.** Simple word embedding based model with modified hierarchical pooling strategy.

pooling strategies operated on top of word embeddings alone could carry sufficient information on distinct NLP tasks in an experimental and interpretative way.

In this paper, a series of pooling strategies are explored in comparative experiments. Specifically, a modified hierarchical operation is proposed to extract critical word as well as $n$-gram spatial information, which demonstrates superior performance for long text classification among several alternative methods.

## III. METHODS

Consider a text sequence $X$ (either a sentence or a document) composed of a sequence of tokens $\{w_1, w_2, \cdots, w_L\}$, where $L$ is the number of tokens, namely, the sentence or document length. Let $\{v_1, v_2, \cdots, v_L\}$ denote the corresponding word embeddings for each token, where $v_i \in R^K$, $K$ is the dimension of the word embeddings. The compositional function, $X \rightarrow z$, aims to combine sequential word embeddings into a fixed-length sentence/document representation $z$. These representations are then used for further tasks such as classification that makes predictions about sequence $X$. The pipeline of the proposed method is shown in Figure 2 and the consisting modules will be introduced in detail below.

### A. WORD EMBEDDINGS

Leveraging word embeddings that were pre-trained with an unsupervised neural language model is a popular and effective method in the absence of a large special supervised training set. For English word embedding, we use the publicly available word2vec vectors $GoogleNews - vectors - negative300$ that were trained on 100 billion words from Google News corpus. The vectors are of 300-dimension and were trained using the continuous bag-of-words architecture.

For Chinese word embedding, we take a public pre-trained word2vec word embedding model [23] that was initialized as 256 dimensions. With skip-gram structure, the model is trained on more than 8 million of public articles from Wechat platform. Before word embedding, a Chinese words segmentation toolbox named *jieba* was utilized to segment a sequential text into individual words. Specially, the Out-Of-Vocabulary (OOV) words that are not present in the set of pre-trained corpus are initialized as zero vectors during the experimental implementation.

### B. SIMPLE WORD EMBEDDING BASED MODELS

Due to the limitation of data deficiency in the few-shot learning tasks, we consider a series of models with no additional compositional parameters to encode natural language sequences, termed Simple Word Embedding based Models (SWEMs).

#### 1) MEAN POOLING

The simplest strategy to form a sentence representation from sequential word embeddings is to compute the element-wise average over given word vectors sequence, as used in DAN [4] and fastText [5]:

$$z^{mean} = \frac{1}{L}\sum_{i=1}^{L} v_i \tag{1}$$

The mean pooling operation takes the average over each of the $K$ dimensions for all word embeddings, resulting in a mean pooling representation $z^{mean}$ with the same dimension as the word embeddings. Intuitively, the mean pooling strategy takes every sequence element into account via simple average operation and treats all word embeddings on equal terms.

#### 2) MAX POOLING

Due to the fact that in general only some key words in a sentence/document contribute to final predictions, another pooling strategy that extracts the most salient features from each word embedding dimension was proposed as:

$$z_j^{max} = \max_{i=1}^{L} v_{ij} \tag{2}$$

where $v_{ij}$ and $z_j^{max}$ are the $j$-th component of the word embedding $v_i$ and the final global max pooling representation $z^{max}$, respectively.

With this pooling strategy, the words that are unimportant or unrelated to the document topic will be filtered during the encoding process as those irrelevant components in the

distributed word vectors will have small amplitude, different from the mean pooling where each word will contribute to the final representation.

#### 3) CONCATENATED POOLING

To some extent, the mean pooling and max pooling operations are complementary in the sense of accounting for different types of information from text sequences. In this case, a combination method that concatenates the mean pooling and max pooling features together was proposed to form the final sentence embedding, termed as concatenated pooling here.

$$z^{concat} = [z^{mean}, z^{max}] \tag{3}$$

#### 4) HIERARCHICAL POOLING

Both the mean pooling and max pooling strategy do not take the word-order or spatial information into account, which could be critical and useful for some special NLP tasks, such as the sentiment classification [11]. By utilizing the local spatial information, a hierarchical pooling strategy was proposed in [11]. Given a sequence of word embeddings $v = \{v_1, v_2, \cdots, v_L\}$, define the $v_{i:i+n-1}$ as a local window consisting of $n$ consecutive words, $v_i$, $v_{i+1}$, $\cdots$, $v_{i+n-1}$. First, a mean pooling strategy is performed on stride-sliding local windows, $v_{1:n}$, $v_{1+s:n+s}$, $v_{1+2s:n+2s}$,$\cdots$ until the last one $v_{L-n+1:L}$ where $n$ is the local window size and $s$ is the stride just like the convolution kernels in CNNs. After mean pooling, the extracted intermediate features from all windows are further aggregated with a global max pooling operation to form the final sentence representation. Due to its layered pooling of mean and max operations, this approach is called hierarchical pooling [11].

#### 5) MODIFIED HIERARCHICAL POOLING

Considering the complementary advantages of the mean and max pooling operations and inspired by the hierarchical structure, we propose a modified hierarchical pooling strategy. As shown in Figure 2, we first extract local critical information in sliding windows with max pooling operation, and then the intermediate features obtained are integrated into a mean pooling layer, leading to a global representation for the whole sentence or document. It could be seen as reversing the implementation order of the mean and max operations in hierarchical pooling strategy. Different from the hierarchical pooling strategy above, the prior max pooling in modified strategy will filter out the irrelevant terms in the local windows. Meanwhile, it will preserve the local spatial information of a text sequence in the sense that it keeps track of how the sentence/document is constructed form individual windows, just like $n$-grams, which may be beneficial to semantic text representation.

For all the pooling strategies above, there are no additional compositional parameters involved to be learned and thus the models only exploit intrinsic word embedding information forwardly without any training phases.

**TABLE 1. Datasets statistics. #c and #w denote the number of classes and average number of tokens in text samples, respectively. Train and test numbers are the original data splits for corresponding datasets used for deep models training and evaluating.**

| Datasets | Language | #c | #w | Train | Test |
|---|---|---|---|---|---|
| Netease | Chinese | 6 | 582 | 24K | |
| Cnews | Chinese | 10 | 530 | 50K | 10K |
| AG News | English | 4 | 43 | 120K | 7.6K |
| DBpedia | English | 14 | 57 | 560K | 70K |
| Yahoo | English | 10 | 104 | 1400K | 60K |

### C. SVM CLASSIFICATION

By feeding an input text sequence into the SWEMs, we directly compute a fixed-dimension feature vector (256-dimension for Chinese and 300-dimension for English text) in a feed-forward way and consider the vector as a global representation of the input text. After that the extracted feature vectors of the several supporting examples are taken to train a linear SVM classifier which is further used for category prediction for the query examples.

## IV. EXPERIMENTS

We evaluate the simple word embedding based methods and compare with other alternative approaches on a variety of text classification datasets. The detailed experimental setup and numerous quantitative analysis as well as visualization are also presented.

### A. DATASETS AND SETTINGS

The few-shot transfer learning experiments are implemented on 5 natural language text classification datasets with the data statistics summarized in Table 1.

Netease and Cnews are two public Chinese text classification datasets. There are 6 categories and only 4000 samples for each category in Netease. Cnews is a subset of the THUCNews dataset, a Chinese text classification dataset produced by Natural Language Processing and Computational Social Science Lab, Tsinghua University. There are 10 categories with 5000/500/1000 samples each category for train/validation/test. The samples in Netease and Cnews are long text documents with 582 and 530 tokens on average, respectively.

About the English text datasets [9], AG News and Yahoo are two topic categorization datasets with 4 and 10 categories, respectively. The average word numbers are 43 for AG News and 104 for Yahoo. DBpedia is an ontology classification dataset with 14 categories and 57 words each sample on average.

For few-shot transfer learning, $M$ sampled classes are firstly selected and then $N$ shot samples and $Q$ query samples are randomly selected from each of the sampled classes, which is called $M$-way $N$-shot $Q$-query. For example, in 5-way 1-shot 15-query settings, there are 1 support and 15 testing samples for each of the 5 sampled classes in each evaluation episode. Following the standard settings adopted by most existing few-shot learning works, we respectively

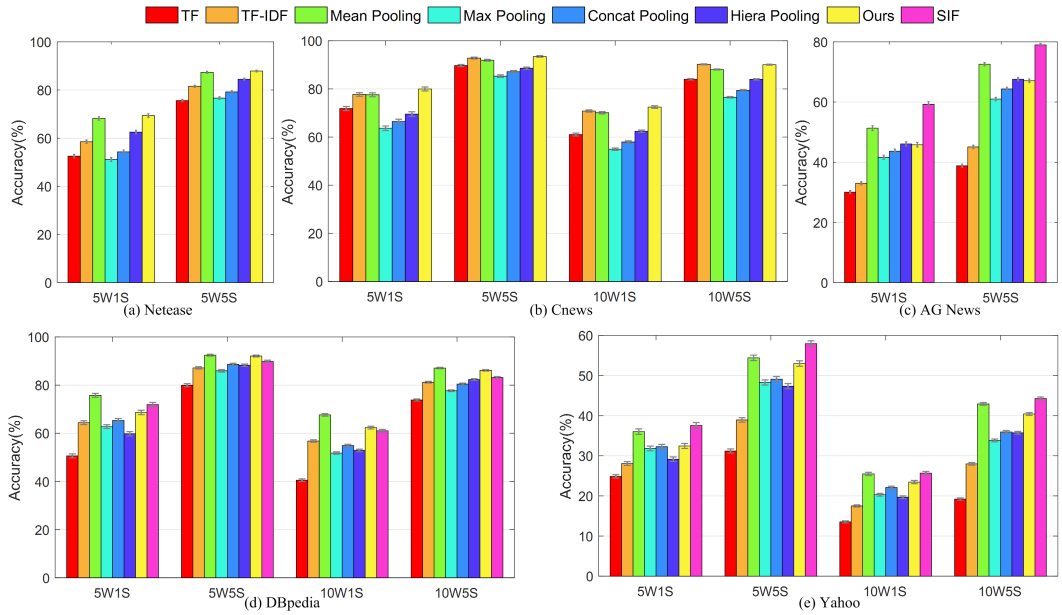



**FIGURE 3.** Few-shot classification accuracy on various datasets. The 95% confidence intervals are attached along with the corresponding accuracy.

conduct 5-way and 10-way, 1-shot and 5-shot classification if possible (4-way classification for AG News due to the 4 categories in total) and batch 15 query samples per class in each episode. The classification accuracy in each episode is measured by $N_c/N_t$ where $N_c$ and $N_t$ denote the number of correctly classified samples and the total number of the testing samples, respectively. The final classification performance is evaluated by averaging over 600 randomly generated episodes from the test set (each episode with randomly selected support and testing samples). Notably, our method is based on the word embeddings that were pre-trained on freely available resources. For all the 5 text classification datasets, only the *test* sets are used throughout the experiments. In summary, the models transfer knowledge from a common source domain to these special target domains.

Term Frequency (TF) and Term Frequency-Inverse Document Frequency (TF-IDF), two widely used Bag-of-Word (BOW) features, are employed as baseline methods for few-shot classification. The TF and TF-IDF features are extracted with the *T fidfV ectorizer* and *CountV ectorizer* modules in the public *sklearn* library during experimental implementation. The public *sklearn* library in Python is used for SVM training and testing with the *LinearSVC* module. In addition, some SWEMs proposed in [11] are implemented for comparison, including the mean pooling, max pooling, concatenated pooling, and hierarchical pooling. What is more, the recent few-shot text classification method SIF in [21], [22] is also investigated and we carry out comparative experiments using publicly available codes from the original publication. However, due to the dependence on auxiliary word frequency information, the SIF method is now only available to English text.

We carry out the experiments on a server equipped with 12-Core Intel i7-8700K CPU and GeForce GTX 1080 Ti GPU, 64 GB of RAM. The Ubuntu 16.04 is used as software environment and the codes are written in Python.

### B. FEW-SHOT TRANSFER LEARNING

We begin with the text classification task in the few-shot transfer learning way and the results are shown in Table 2 and Figure 3. For simplicity, we only list the comparative results for 2 representative datasets Cnews (Chinese long text) and DBpedia (English short text) in Table 2 while all the classification accuracies on 5 datasets are presented with a set of bar charts as shown in Figure 3.

We present the few-shot classification results in regard of two kinds of datasets, long text Chinese documents datasets and short text English documents datasets. For Chinese text datasets of Netease (Figure 3 (a)) and Cnews (Figure 3 (b)), the first observation is that our proposed modified hierarchical pooling method performs best in most cases except for the comparable performance with TF-IDF on Cnews 10-way 5-shot task. More specifically, mean pooling is the next best method on Netease while TF-IDF, as well as the mean pooling, achieves slightly inferior performances than our method on Cnews dataset. Among all these methods, max pooling achieves almost bottom-ranked performances due to the ignorance of local critical information, indicating that the word-order spatial patterns of a text sequence are relatively advantageous for semantic representation. For English short text datasets, including AG News (Figure 3 (c)), DBpedia (Figure 3 (d)), and Yahoo (Figure 3 (e)), the simple mean pooling and SIF methods achieve significant performances towards the few-shot classification tasks. An interesting

**TABLE 2.** Few-shot classification accuracy on Cnews and DBpedia datasets. All accuracy results are averaged over 600 test episodes and are reported with 95% confidence intervals, same as [16]. For each task, the best-performing method is highlighted, along with any others whose confidence intervals overlap.

| | Cnews | | | | DBpedia | | | |
|---|---|---|---|---|---|---|---|---|
| | 5-way Acc.(%) | | 10-way Acc.(%) | | 5-way Acc.(%) | | 10-way Acc.(%) | |
| | 1-shot | 5-shot | 1-shot | 5-shot | 1-shot | 5-shot | 1-shot | 5-shot |
| TF | 71.85±0.86 | 89.70±0.43 | 61.04±0.60 | 84.02±0.30 | 50.61±0.82 | 79.97±0.65 | 40.53±0.53 | 73.83±0.41 |
| TF-IDF | 77.68±0.74 | 92.81±0.38 | 70.83±0.48 | **90.21±0.21** | 64.42±0.79 | 87.16±0.49 | 56.79±0.52 | 81.22±0.35 |
| Mean Pooling [11] | 77.58±0.82 | 91.90±0.39 | 70.11±0.48 | 88.08±0.22 | **75.75±0.81** | **92.45±0.39** | **67.64±0.53** | **87.10±0.30** |
| Max Pooling [11] | 63.67±0.94 | 85.30±0.55 | 54.95±0.50 | 76.54±0.30 | 62.77±0.78 | 85.95±0.52 | 51.78±0.51 | 77.67±0.36 |
| Concat Pooling [11] | 66.47±0.95 | 87.15±0.52 | 57.97±0.49 | 79.43±0.29 | 65.32±0.81 | 88.64±0.46 | 54.92±0.50 | 80.46±0.37 |
| Hiera Pooling [11] | 69.57±0.93 | 88.60±0.46 | 62.41±0.51 | 83.98±0.28 | 59.79±0.90 | 88.25±0.49 | 52.88±0.51 | 82.32±0.34 |
| SIF [21] | - | - | - | - | 71.90±0.91 | 89.86±0.50 | 61.03±0.59 | 83.20±0.36 |
| Ours | **79.79±0.82** | **93.49±0.33** | **73.00±0.51** | **90.07±0.20** | 68.66±0.92 | 92.10±0.37 | 62.35±0.58 | 86.13±0.31 |

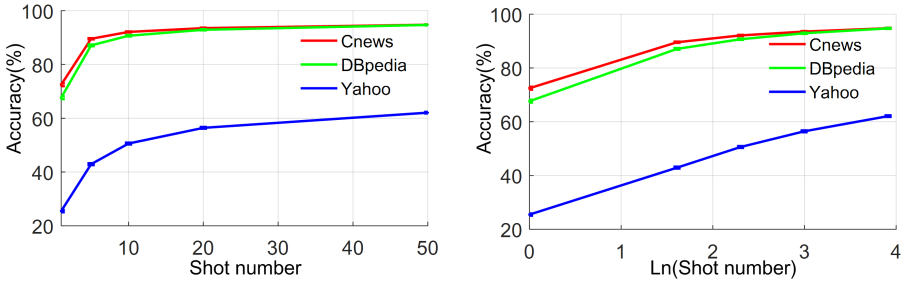



**FIGURE 4.** The increasing accuracy with more supporting samples for 10-way classification task on different datasets.

phenomenon is that the SIF method performs better than others on the few-category datasets such as AG News and Yahoo while achieves lower accuracies than mean pooling and our method on DBpedia dataset which has relatively more categories, as indicated in the original publication [21] that it is best suited to few-category classification tasks.

At a holistic level, the simple mean pooling methods and its derivations are well suited to short text while our modified hierarchical pooling method performs better for long text representation, revealing some general rules for rationally selecting models to tackle different tasks. A possible reason is that long text documents contain much more local $n$-gram spatial information and such word-order local patterns may be discriminative for predicting the content of a document and more easily captured by the modified hierarchical pooling method. Finally, it is worth to note that although derived from the hierarchical pooling strategy proposed in [11], our proposed modified one achieves higher accuracies on almost all datasets, including both Chinese and English text.

To investigate the computational efficiency, we summary the dynamic hardware resource requirement as well as the running time for each comparative method, as illustrated in Table 3 It can be seen that the SWEMs with different pooling strategies share nearly similar resource occupation and running time, including the mean pooling, max pooling, concatenated pooling, hierarchical pooling, and our modified hierarchical pooling methods. Compared to SWEMs, the baseline methods TF and TF-IDF will take more CPU resources and lead to less running time. In contrast, the recent SIF method poses a higher demand for both CPU and memory occupation and it even takes about twice the running time than our method.

**TABLE 3.** Computational resources statistics on DBpedia (10-way 5-shot). The maximum CPU (multi-core) and memory utilizations during processing are provided in %. The running time is recorded for a total evaluation with 600 randomly generated episodes.

| Models | CPU(%) | Memory(%) | Running Time(s) |
|---|---|---|---|
| TF | 663.3 | 4.1 | 86 |
| TF-IDF | 654.5 | 4.1 | 86 |
| Mean Pooling | 413.3 | 4.2 | 161 |
| Max Pooling | 388.3 | 4.2 | 172 |
| Concat Pooling | 362.7 | 4.2 | 194 |
| Hiera Pooling | 359.7 | 4.2 | 182 |
| SIF | 797.7 | 11.3 | 383 |
| Ours | 359.7 | 4.2 | 192 |

### C. MORE SHOT SAMPLES

Current deep learning models need a huge number of examples to tune and adjust the thousands of parameters. However, it is infeasible in few-shot transfer learning tasks due to insufficient data. To explore the contribution of more supporting examples in few-shot classification, we investigate the classification accuracies with growing number of the supporting shot examples and the results are shown in Figure 4 (a). Notably, the accuracy goes up sharply with increasing shot examples at the beginning and tends to reach saturation at a certain turning point, such as 10 samples for Cnews and

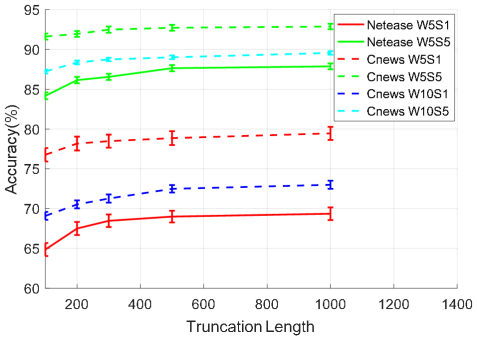



**FIGURE 5.** The few-shot classification accuracy as a function of the truncation length of the input text.

DBpedia datasets. To better illustrate, we map the relationship between accuracy and the shot sample number in logarithmic coordinates as shown in Figure 4 (b), where the approximate linear characteristics of the polylines indicate the nearly logarithmic relation between the accuracy and shot number. Conclusions could be drawn that a turning-point number of supporting examples are appropriate for few-shot classification and more samples will lead to slight performance enhancement and exponentially growing data cost. Finally, it is obvious that the classification performance on Yahoo dataset is far behind than that on Cnews and DBpedia. It could be explained that the Yahoo dataset is organized in conversation form that includes question contexts and best answers instead of the declarative sentences or descriptive documents in Cnews and DBpedia. Another reason may be that the topic categories in Yahoo are semantically similar and hence lead to ambiguity in the few-shot classification task.

### D. EFFECT OF TEXT TRUNCATION

The results reported above are obtained with the whole length of the input text sequences. However, in some models like CNNs, the input sequences are truncated to a pre-defined fixed length for further encoding, i.e., the tokens beyond the truncation length will be discarded. To verify the effectiveness of this empirical setting, we evaluate the classification performances with different truncation lengths and the results are presented in Figure 5. Expectedly, the classification accuracy keeps going up with the increasing truncation length due to the relatively more information about the inputs. Another interesting finding is that partly down-sampling the inputs just leads to slight performance dropping, for instance, the accuracy drops from 79.46±0.82% to 78.48±0.82% with the truncation length changed from 1000 to 300 for 5-way 1-shot classification on Cnews dataset (the average token number is 530). Hence, we conclude that a tradeoff between accuracy and efficiency could be made by appropriate truncation of the input text although more information yields better performance.

### E. SLIDING WINDOW PARAMETERS ANALYZATION

To investigate the influence of the sliding window parameters (local window size and stride) on the few-shot classification

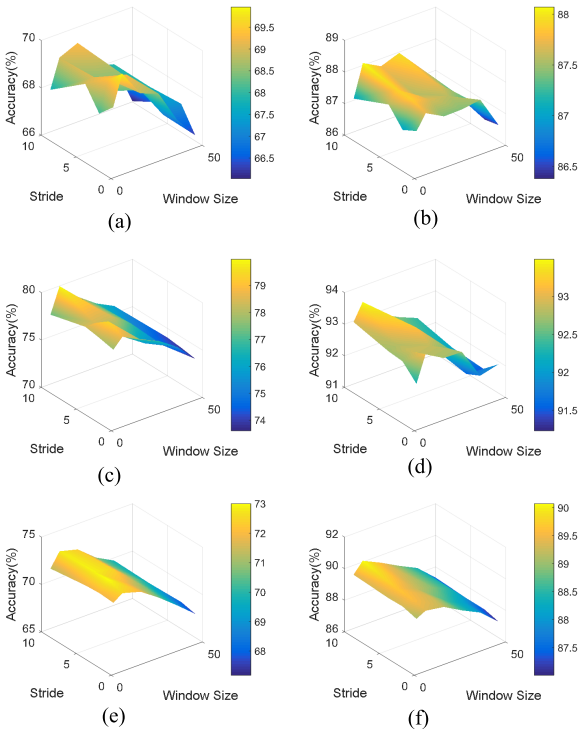



**FIGURE 6.** The classification accuracies with different sliding window parameters. (a) 5-way 1-shot for Netease dataset. (b) 5-way 5-shot for Netease dataset. (c) 5-way 1-shot for Cnews dataset. (d) 5-way 5-shot for Cnews dataset. (e) 10-way 1-shot for Cnews dataset. (f) 10-way 5-shot for Cnews dataset.

performance and the robustness of the proposed modified hierarchical pooling strategy, we conduct extensive experiments by traversing a 2D parameter grid, i.e., window size selected over {5, 10, 20, 30, 40, 50} and stride selected over {1, 3, 5, 10}. The classification model with modified hierarchical pooling strategy is implemented on the two long text datasets Netease and Cnews and the results are shown in Figure 6. Approximately, the accuracy changes in a reasonably small range and the gradients are more significant with the variation of the window size, forming a peak at around 10. The stride in sliding window plays a relatively trivial role in the classification task especially for Cnews dataset. A conclusion could be drawn that the performance is relatively stable and regular, making it easy to find an optimal value for parameters setting.

### F. VISUALIZATION AND EXPLANATION

In order to explain the effectiveness of the SWEMs towards the text representation and classification, we visualize the text features encoded via the simple word embedding based methods, including the mean pooling strategy for the short text documents and modified hierarchical pooling for the long ones. The extracted high-dimensional feature vectors of the text documents are visualized in a 2D space by using the t-SNE algorithm [24]. As shown in Figure 7, each number in the graphs represents the 2D feature of a labeled text example and each color indicates a different category. Obviously,

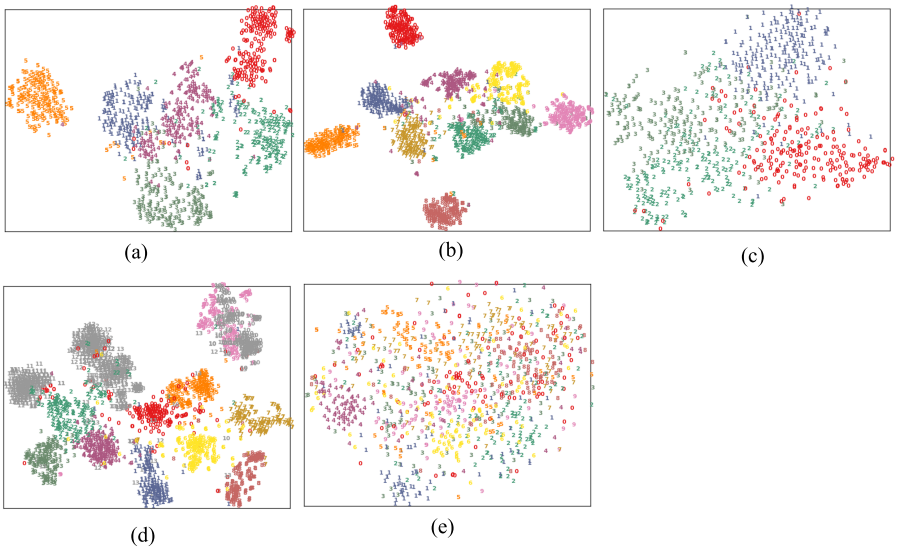



**FIGURE 7.** 2D feature visualization of the SWEMs text representation. (a) Netease dataset. (b) CNews dataset. (c) AG News dataset. (d) DBpedia dataset. (e) Yohoo dataset.

**TABLE 4.** Comparisons of CNN, LSTM and SWEM architectures. *L* and *K* denote the tokens number in the sentences/documents and word embeddings dimension, respectively. Here, *n* denotes the filter width for the CNN and *d* represents the dimension of hidden units in LSTM or the number of filters in CNN [25].

| Models | Parameters | Complexity | Sequential Operations |
|---|---|---|---|
| CNN | $n*K*d$ | o($n*L*K*d$) | o(1) |
| LSTM | $4*d*(K+d)$ | o($L*d^2+L*K*d$) | o($L$) |
| SWEM | 0 | o($L*K$) | o(1) |

the 2D features in Figure 7 (a), (b), (c), and (d) naturally tend to form clusters that are clearly separated. However, due to the semantically similar categories of the Yahoo dataset, the 2D features in Figure 7 (e) only form a few visible clusters and many of them overlap with each other in a desultory manner, explaining the relatively inferior classification accuracy compared to other datasets. The strong results of the SWEM approaches on these datasets demonstrate that it is sufficient in most cases to simply model the homogeneous word-level and even hierarchical word-order spatial information in a text sequence.

## V. DISCUSSION

### A. PARAMETERS AND COMPUTATION COMPARISON

It is worth mentioning that the SWEMs discussed in this paper have no more compositional parameters involved and no iteratively training process, which is critical and necessary for few-shot transfer learning where no large amounts of supervised data exists. To take a quantitative comparison, the compositional parameters, computational complexity, and sequential operations are summarized in Table 4. Both the CNN and LSTM have a huge number of parameters to fit the semantic compositionality of text sequences while SWEM has none. In term of the computational complexity, SWEM tends to be more efficient than CNN and LSTM by a magnitude factor of $nd$ or $d$.

### B. THE SOLID FOUNDATION OF THE PRE-TRAINED DISTRIBUTED WORD VECTOR REPRESENTATION

Word embeddings, learned from massive unsupervised text corpus, are widely-used fundamental building blocks in NLP tasks. By representing each word as a fixed-length vector, these embeddings could form semantically similar clusters, while implicitly encoding rich linguistic regularities and patterns [2], [3]. Extensive experimental investigations have validated that simple pooling strategies operated over word embeddings alone already carry sufficient information for natural language understanding [11].

### C. IMPORTANCE OF WORD-ORDER INFORMATION

One possible disadvantage of the pure max pooling or mean pooling operation is that they neglect the word-order information within a text sequence, which is significant for sentiment analysis tasks and would be naturally captured by CNN or LSTM based deep models. However, an empirical research [11] has revealed that the hierarchical pooling strategy manages to abstract word-order and spatial information from the input sequence by incorporating the local window information, i.e., $n$-gram features.

### D. THE DIFFERENCE BETWEEN HIERARCHICAL POOLING STRATEGY AND THE PROPOSED MODIFIED ONE

Considering a particular consumer review in a sentiment analysis task '*I* ***like*** *the food in this restaurant and the environment is* ***good***', the purpose is to identify how much the consumer like or hate the products, i.e., the positive or negative sentiment attitude based on the text comments. We are going to assign the sentiment component in the distributed embedding vectors according to semantic comprehension, such as assigning '*like*' as 1.0 and '*dislike*' as −1.0. Without loss of generality, the components of the tokens in this review could be listed as {0.1,**1.0**, −0.1, 0.1, 0, 0.1, −0.1, 0.0, −0.1, 0.1, −0.1, **0.9**} where the critical words '*like*' and '*good*'

are assigned as 1.0 and 0.9, respectively. Considering the hierarchical pooling operation with local window size and stride both set as 4, the final representation is calculated as max{mean(0.1,**1.0**, −0.1, 0.1), mean(0, 0.1, −0.1, 0), mean(−0.1, 0.1, −0.1,**0.9**)} = 0.275 which only contains the information of the key word '*like*', regardless of another key word '*good*', resulting in a inadequate positive score 0.275. In contrast, with the modified hierarchical pooling operation, the sentiment score is calculated as mean{max(0.1,**1.0**, −0.1, 0.1), max (0, 0.1, −0.1, 0), max (−0.1, 0.1, −0.1,**0.9**)} ≈ 0.667. With the critical words '*like*' and '*good*' information involved both, the modified hierarchical pooling strategy leads to a much better positive score than the original one.

## VI. CONCLUSION

In this paper, we propose a modified hierarchical pooling strategy for simple word embedding based models for text classification in the few-shot transfer learning way. Extensive experiments on 5 NLP datasets indicate that the lightweight SWEMs are effective and efficient enough for text classification, both for English and Chinese text, especially in few labeled examples supporting situations. Furthermore, the simple mean pooling strategy is adequate to represent and classify the short text documents. However, for long text documents with hundreds of words/tokens, the proposed modified hierarchical pooling strategy performs better due to the consideration of the word-order and local spatial information. It is also indicated that the features extracted from the SWEMs have robust representative ability and may be potential for other NLP tasks, such as sentiment analysis and question answering, which will be left as further open research.

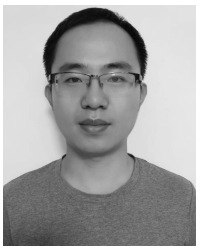
## REFERENCES

[1] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.
[2] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. NIPS*, 2013, pp. 3111–3119.
[3] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proc. EMNLP*, 2014, pp. 1532–1543.
[4] M. Iyyer, V. Manjunatha, J. Boyd-Graber, and H. Daumé, III, "Deep unordered composition rivals syntactic methods for text classification," in *Proc. IJCNLP*, 2015, pp. 1681–1691.
[5] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," in *Proc. EACL*, 2017, pp. 427–431.
[6] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. NIPS*, 2014, pp. 3104–3112.
[7] K. S. Tai, R. Socher, and C. D. Manning, "Improved semantic representations from tree-structured long short-term memory networks," in *Proc. IJCNLP*, 2015, pp. 1556–1566.
[8] Y. Kim, "Convolutional neural networks for sentence classification," in *Proc. EMNLP*, vol. 2014, pp. 1746–1751.
[9] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *Proc. NIPS*, 2015, pp. 649–657.
[10] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," in *Proc. ACL*, 2014, pp. 655–665.
[11] D. Shen *et al.*, "Baseline needs more love: On simple word-embedding-based models and associated pooling mechanisms," in *Proc. ACL*, 2018, pp. 440–450.
[12] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proc. ICML*, 2017, pp. 1126–1135.
[13] S. Ravi and H. Larochelle, "Optimization as a model for few-shot learning," in *Proc. ICLR*, 2017, pp. 1–11.
[14] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *Proc. ICML*, 2015, pp. 1–8.
[15] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra, "Matching networks for one shot learning," in *Proc. NIPS*, 2016, pp. 3630–3638.
[16] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *Proc. NIPS*, 2017, pp. 4077–4087.
[17] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. S. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," in *Proc. CVPR*, 2018, pp. 1199–1208.
[18] L. Yan, Y. Zheng, and J. Cao, "Few-shot learning for short text classification," *Multimedia Tools Appl.*, vol. 77, no. 22, pp. 29799–29810, 2018.
[19] A. P. Parikh, O. Täckström, D. Das, and J. Uszkoreit, "A decomposable attention model for natural language inference," in *Proc. EMNLP*, 2016, pp. 2249–2255.
[20] A. Vaswani *et al.*, "Attention is all you need," in *Proc. NIPS*, 2017, pp. 5998–6008.
[21] K. Bailey and S. Chopra. (2018). "Few-shot text classification with pre-trained word embeddings and a human in the loop." [Online]. Available: https://arxiv.org/abs/1804.02063
[22] S. Arora, Y. Liang, and T. Ma, "A simple but tough-to-beat baseline for sentence embeddings," in *Proc. ICLR*, 2017, pp. 1–16.
[23] *Incredible Word2Vec: A Pre-trained Model*. Accessed: Apr. 3, 2017. [Online]. Available: https://spaces.ac.cn/archives/4304
[24] L. V. Der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.
[25] E. Tola, V. Lepetit, and P. Fua, "DAISY: An efficient dense descriptor applied to wide-baseline stereo," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 5, pp. 815–830, May 2010.

**CHONGYU PAN** received the B.S. degree in mechatronic engineering and automation from the National University of Defense Technology, Changsha, China, in 2014, and the M.S. degree in mechatronic engineering and automation, in 2016. He is currently pursuing the Ph.D. degree in artificial intelligence. His research interests include the control of the unmanned aircraft and the development of the deep learning, especially the few-shot learning and multi-modal learning.

**JIAN HUANG** received the Ph.D. degree in simulation engineering from the National University of Defense Technology, China, in 2000. She is currently a Professor with the College of Intelligence Science. Her current research interests include mission planning and large-scale distributed simulation.

**JIANXING GONG** received the Ph.D. degree in simulation engineering from the National University of Defense Technology, China, in 2007. He is currently an Associate Professor with the College of Intelligence Science. His research interests include task assignment and distributed parallel simulation.



**XINGSHENG YUAN** received the Ph.D. degree in computer science from the National University of Defense Technology, China, in 2014. He is currently a Lecturer with the College of Intelligence Science. His research interests include digital image processing, deep learning, and knowledge reasoning.