Learning Embeddings for Rare Words Leveraging Internet Search Engine and Spatial Location Relationships

Xiaotao Li, Shujuan You, Yawen Niu, Wai Chen
China Mobile Research Institute, Beijing, China
{lixiaotao, youshujuan, niuyawen}@chinamobile.com
wai.w.chen@gmail.com

Abstract

Word embedding techniques depend heavily on the frequencies of words in the corpus, and are negatively impacted by failures in providing reliable representations for low-frequency words or unseen words during training. To address this problem, we propose an algorithm to learn embeddings for rare words based on an Internet search engine and the spatial location relationships. Our algorithm proceeds in two steps. We firstly retrieve webpages corresponding to the rare word through the search engine and parse the returned results to extract a set of most related words. We average the vectors of the related words as the initial vector of the rare word. Then, the location of the rare word in the vector space is iteratively fine-tuned according to the order of its relevances to the related words. Compared to other approaches, our algorithm can learn more accurate representations for a wider range of vocabulary. We evaluate our learned rare-word embeddings on the word relatedness task, and the experimental results show that our algorithm achieves state-of-the-art performance.

1 Introduction

Since Bengio et al. (2003) introduced the idea of learning continuous vectors for words using network-based language models, many word embedding techniques have been proposed such as Word2vec (Mikolov et al., 2013a,b), GloVe (Pennington et al., 2014), etc. However, nearly all existing word embedding approaches need words that have a high frequency in the corpus and cannot learn good representations for rare words (including low-frequency words and unseen words). As words in a corpus follow a Zipfian distribution, only a small proportion of the total tokens are frequent words, while most of them are rare words. Therefore, how to learn qualified embeddings for rare words is an essential issue to be solved.

From the human perspective, when encountering a new word, it is an instinct to take a look at its structure or to look up its definition in a dictionary. The essence of both behaviours is to transform a rare word to a set of familiar words expressing the same meaning to it. Based on the above ideas, some proposed techniques have attempted to exploit subword information or lexical resources to predict the rare word representation.

In the area of subword-based approaches, FastText (Bojanowski et al., 2017) learns representations for character n-grams and represents words as the sum of the n-gram vectors. Ngram2vec (Zhao et al., 2017) learns n-gram representations from n-gram co-occurrence statistics and incorporates this information into the word representations. Pinter et al. (2017) proposed the Mimick model to predict vectors for out-of-vocabulary words by learning a function from spellings to distributional embeddings. The attentive mimicking model (AM) (Schick and Schütze, 2019a) and the form context model (FCM) (Schick and Schütze, 2019b) jointly use surface form and context information to improve representations of rare words.

In another way, lexical resources are used to infer the representation for a rare word from the vectors of the words having a semantic association with it. SemLand (Pilehvar and Collier., 2017) infers the representations for rare words by exploiting the definitions and relationships in an external lexicon WordNet (Miller, 1995). Faruqui et al. (2015) proposed to use word relation knowledge found in semantic lexicons to retrofit word vectors. Bahdanau et al. (2018) proposed to train a Long Short-Term Memory (LSTM) network to predict the representations of rare words based on auxiliary data (e.g., a dictionary definition) from knowledge bases. Prokhorov et al. (2019) embedded a knowledge base into a vector space by the node2vec (Grover and Leskovec, 2016) graph embedding algorithm and then mapped the embedded words from this space to a corpus-based space. However, the performance of these approaches

Proceedings of the 10th Conference on Lexical and Computational Semantics, pages 278–287
August 5–6, 2021, Bangkok, Thailand (online) ©2021 Association for Computational Linguistics

heavily depends on the coverage of external data sources. If a rare word is uncovered by the lexicon, the rare-word embedding will not be available.

In addition to the approaches outlined above, a great concern has been raised over the pre-trained language models for their outstanding performance in various natural language processing (NLP) tasks. Among the pre-trained models, ELMo (Peters et al., 2018) and BERT (Devlin et al., 2019) are two most typical ones. Based on the pre-trained language model, we can use a function of the internal layers as the vector of a word. The pre-trained language models have strong coverage ability and can predict vectors for nearly all rare words. However, based on our experimental results (section 3.4), we found the semantics of rare words are not learned well when the context information is not provided or the rare words come from specific domains.

As the largest source of information in the world, the Internet consists of billions of pages of data in all fields. To figure out the meaning of a rare word, almost everyone's first priority has changed to retrieve it on the Internet and extract the useful information from the associated webpages. Inspired by this, we propose a two-step algorithm to learn rare-word embeddings using the Internet search engine and the spatial location relationships. We firstly find the top-n most relevant words to a rare word from the webpages returned by the Internet search engine and compute the initial embedding of the rare word by averaging the vectors of these extracted words. According to the order of the top-n most relevant words, we further iteratively fine-tune the location of the rare word in the vector space to make it satisfy the constraints of spatial location relationships. The constraints are that if a rare word is more relevant to a word than other words, the distance between the rare word and this word is closer than the distances between the rare word to others in the vector space. Compared to the existing approaches, there are three advantages of our approach: (i) we can obtain a powerful coverage for rare words; (ii) we can provide more accurate vector representations for rare words; (iii) we can support representing multilingual rare words.

This paper is organized as follows: Section 2 describes our methodology in detail. Section 3 presents the experimental results. The paper is concluded in Section 4.

2 Methodology

In this section, we will begin by introducing our motivation, then describe how we define the relevance metric and obtain the top-n related words to a rare word using the Internet search engine, and finally present the fine-tuning process toward achieving the more precise embedding learning.

2.1 Motivation

To solve the rare word representation problem, as mentioned above, the most direct way is to find a series of familiar tokens expressing the same meaning to the rare word. Further, the embedding of a rare word can be induced by the embedding of its semantically related tokens in the word embedding model. Based on the above analysis, there are two main challenges in the task of the rare word representation: (i) how to obtain the semantically related words for more rare words? and (ii) how to ensure the quality of the learned rare-word embeddings? To address these issues, we propose an algorithm to learn embeddings for rare words, as shown in Figure 1, which consists of two processes: a coarse-tuning one and a fine-tuning one. The coarse-tuning process is to obtain the semantically related words for a rare word and to predict its approximate location in the vector space (i.e., the coarse-grained representation). The fine-tuning process is to adjust the coarse-grained vector of the rare word intensively to optimize its meaning representation accuracy; and the final learned-vector is the fine-grained representation.

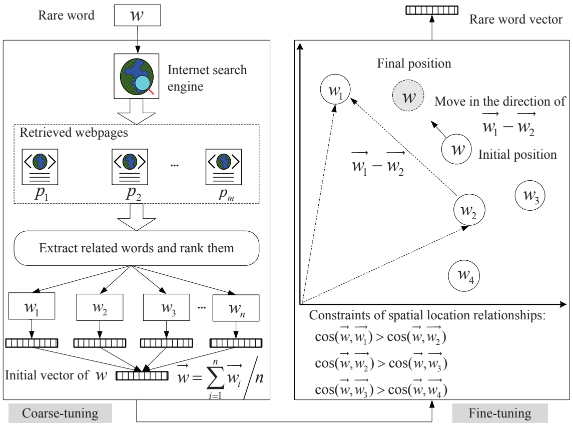
Rare word
w
Internet search engine
Retrieved webpages
p_1
p_2
...
p_m
Extract related words and rank them
w_1
w_2
w_3
...
w_n
Initial vector of w
$\vec{w} = \sum_{i=1}^{n} \vec{w_i} / n$
Coarse-tuning
Rare word vector
Final position
w
Move in the direction of $\vec{w_1} - \vec{w_2}$
w
Initial position
$\vec{w_1} - \vec{w_2}$
w_1
w_2
w_3
w_4
Constraints of spatial location relationships:
$\cos(\vec{w}, \vec{w_1}) > \cos(\vec{w}, \vec{w_2})$
$\cos(\vec{w}, \vec{w_2}) > \cos(\vec{w}, \vec{w_3})$
$\cos(\vec{w}, \vec{w_3}) > \cos(\vec{w}, \vec{w_4})$
Fine-tuning

Figure 1: Procedure of learning rare-word embeddings.

To extend the coverage for rare words in the coarse-tuning process, we use information on the Internet as the data source and utilize the Internet search engine to achieve fast acquisition of the topic-related webpages to the rare word. With the help of the large-scale data on the Internet, we can find the semantically related information for nearly all of the words in different fields, which brings better coverage capacity than the existing lexicon-based approaches. As the webpage documents vary widely in their internal structure and contain a lot of interference information such as advertisements, navigation texts, etc, we use only the titles from the retrieved documents to extract the semantically related words of the rare word.

For a rare word w, we firstly use a search engine to search for w and capture the first m relevant records: $P = \{p_1, p_2, p_3, ..., p_m\}$. Then, a set of n most related words $W = \{w_1, w_2, w_3, ..., w_n\}$ to the rare word are extracted from the titles of these m relevant webpages and these n words are ranked by relevance from high to low. The vectors of these n related words is averaged as the initial vector of the rare word (see Equation (1)), which is the coarse-grained representation of w.

$$\vec{w} = \frac{\sum_{i=1}^{n} \vec{w_i}}{n}, \tag{1}$$

where $\vec{w_i}$ is the vector representation of the related word $w_i \in W$. To further ascertain the location of the rare word in the vector space, we consider the constraints of spatial location relationships among the rare word and its n semantically related words. As the related words have been ranked by the relevance to the rare word w, the semantic relatedness between the rare word and a high-ranking word in W is higher than that between the rare word and a low-ranking word in W, i.e.:

$$\begin{aligned} & rel(w, w_i) > rel(w, w_j), \\ & \forall i, j \in [1, n], w_i \in W, w_j \in W, i < j, \end{aligned} \tag{2}$$

where rel is a metric function of the semantic relatedness. In the word embedding model, the semantic relatedness between two words can be represented as the cosine distance of their vectors in the vector space. Therefore, the constraints can be expressed as follows:

$$\begin{aligned} & \cos(\vec{w}, \vec{w_i}) > \cos(\vec{w}, \vec{w_j}), \\ & \forall i, j \in [1, n], w_i \in W, w_j \in W, i < j. \end{aligned} \tag{3}$$

To satisfy the constraints of the spatial location relationships, the vector of the rare word is iteratively fine-tuned as follows:

$$\begin{aligned} & \vec{w} = \vec{w} + (\vec{w_i} - \vec{w_j}) \times \Delta, \\ & if : (\cos(\vec{w}, \vec{w_i}) < \cos(\vec{w}, \vec{w_j})) and (i < j), \end{aligned} \tag{4}$$

where $\vec{w_i} - \vec{w_j}$ is the movement direction of w, and Δ is the step length. As shown in Figure 1, the location of the rare word w is the center of its four semantically related words after the coarse-tuning process, where the hyper-parameter n is set to 4 as an example. However, the current coarse-grained vector of w does not satisfy the constraint: $\cos(\vec{w}, \vec{w_1}) > \cos(\vec{w}, \vec{w_2})$. In the fine-tuning process, the location of w gradually heads toward w_1 in the direction of $\vec{w_1} - \vec{w_2}$, and the final location of w is closer to its real location.

2.2 Coarse-grained Rare Word Representation

The specific procedure of the coarse-tuning process is described in Algorithm 1. We define a dictionary of key-value pairs to store the relevance scores among the rare word and its semantic related words (Line 3). Given a rare word w, the search engine S is invoked to query the related webpages P (Line 4). For each page, the `lxml` module of Python is exploited to extract its title. We decompose it into a set of distinct words and delete the stop words from the segmentation result (Lines 8-11). Based on our tests, there is a list of titles lacking of discrimination and interfering the acquisition of related words. Take the English word "self-discipline" for example, the title of one of its related webpages is "*what is self-discipline - definitions*". The word "definitions" is not exclusively related to "self-discipline" in meaning, because this word also appears in the titles of the retrieved webpages for numerous other search words. To address this issue, we define a noise word set Γ which currently includes 10 words: {"**definition**", "**wiktionary**", "**synonyms**", "**antonyms**", "**dictionary**", "**blog**", "**html**", "**www**", " **encyclopedia**", " **journal**"}. If a title contains a word of Γ, it will be abandoned in our algorithm (Line 13).

At this point, the keys of *map* (i.e., semantic related words) are assigned with the words included in the filtered webpage titles. To measure the relevance score between two words, we take the co-occurrence information and the number of word meanings into account. The co-occurrence frequency is defined by the number of titles that contain the related word. Since the vector of a polyseme is actually a compromise of all its meanings, the polyseme is likely to locate far from the rare word in the vector space. According to this con-

Algorithm 1 Coarse-tuning.

```
Input: The word embedding model, $M$; the rare
    word, $w \notin M$; the max number of related
    words, $n$; the Internet search engine, $S$; the
    semantic lexicon $L$; the noise word set $\Gamma$;
Output: The semantically related word set of $w$,
    $W$; the coarse-grained embedding of $w$, $\vec{w_c}$;
 1: Initialize $\vec{w_c} \leftarrow \vec{0}$;
 2: // $key$ is a semantically related word to $w$, and
    $value$ is the relevance score
 3: $map \leftarrow dict(key, value)$;
 4: $P \leftarrow Search(S, w)$;
 5: for each $p \in P$ do
 6:     // Extract the title of the webpage $p$
 7:     $S \leftarrow GetTitle(p)$;
 8:     // Decompose $S$ into some distinct tokens
 9:     $T \leftarrow Decompose(S)$;
10:     // Remove the stop words
11:     $T \leftarrow RemoveStopWords(T)$;
12:     // Exclude the titles including noise words
13:     if $T \cap \Gamma = \varnothing$ then
14:         for each $t \in T$ do
15:             if $t \in M$ and $t \in L$ then
16:                 // Update the relevance score
17:                 $s = 1/GetSenseNum(L, t)$;
18:                 $map[t] \leftarrow map[t] + s$;
19:             end if
20:         end for
21:     end if
22: end for
23: // Rank the semantic words by the relevance
    score from high to low
24: $map.Sort()$;
25: for each $key \in map.keys$ do
26:     // Get the first $n$ words out of $map$ as the
    semantically related word set $W$
27:     if $|W| < n$ then
28:         $W.append(key)$;
29:         $\vec{w_c} \leftarrow \vec{w_c} + M[key]$;
30:     end if
31: end for
32: if $|W| > 0$ then
33:     $\vec{w_c} \leftarrow \vec{w_c}/|W|$;
34: end if
35: return $\vec{w_c}, W$;
```

sideration, we put more emphasis on the univocal words than the polysemes to infer the rare-word embeddings. The relevance score is proportional to the co-occurrence frequency and inversely proportional to the number of word meanings, i.e.:

$$Score(w, v) = \frac{m_v}{GetSenseNum(L, v)}, \qquad (5)$$

where v is a word related to the rare word w; m_v is the number of webpage titles that include the word v; $GetSenseNum$ is an abstract function to obtain the number of meanings of v, and the parameter L is a semantic lexicon used as a sense inventory. For example, if WordNet (Miller, 1995) is used as the lexicon, the function $GetSenseNum$ is to find the number of synsets that a word belongs to. To provide the candidate meanings and the vector representation for each related word, it requires the semantically related words to be covered by the lexicon and the pre-trained word embedding model (Line 12). It should be noted that there is a clear difference between our algorithm and the lexicon-based approaches. We do not need the rare word to be covered by the lexicon but seek to find a list of related words in the lexicon to learn the rare word vector representation. Therefore, our algorithm is not susceptible to the coverage of the semantic lexicon. Uniformly, for each rare word, we use the top-n most related words and average their embeddings as the coarse-grained representation (Lines 24-34). The parameter n is used to limit the number of semantically related words when the size of map is greater than n.

2.3 Fine-grained Rare Word Representation

The fine-tuning process builds on the coarse-tuning process to optimize the rare word vectors. The main idea of the fine-tuning process is that the more related the two words are, the closer their word embeddings locate in the vector space. Based on this, the specific procedure of this process is described in Algorithm 2. On account of the semantically related words in the order of relevance and the coarse-grained embedding of w, the vector $\vec{w_f}$ is iteratively fine-tuned to fulfill the constraints of the spatial location relationships. The hyper-parameter K is used to control the total number of fine-tuning epochs (Line 5). During a fine-tuning epoch, if the relevance score between the rare word w and each semantically related word $w_i \in W$ is less than the relevance score between the rare word w and each semantically related word $w_j \in W$ with lower order than i, as declared by Equation (4), the rare word w will move one step (Δ) to get closer to w_i (Lines 14-15). Finally, the vector of w will be updated to a new position in the vector space, where the meaning of w can be more accurately represented.

Algorithm 2 Fine-tuning.

```
Input: The word embedding model, $M$; the rare word, $w \notin M$; the semantically related word set of $w$, $W$; the coarse-grained embedding of $w$, $\vec{w_c}$; the number of epochs, $K$; the step size, $\Delta$.
Output: The fine-grained embedding of $w$, $\vec{w_f}$;
Initialize $\vec{w_f} \leftarrow \vec{w_c}$;
// The number of the semantically related words
$n = |W|$;
if $n > 1$ then
    for $k = 1$ to $K$ do
        for $i = 1$ to $n - 1$ do
            $w_i \leftarrow W[i-1]$;
            $\vec{w_i} \leftarrow M[w_i]$;
            $rel_i = \cos(\vec{w_f}, \vec{w_i})$;
            for $j = i + 1$ to $n$ do
                $w_j \leftarrow W[j-1]$;
                $\vec{w_j} \leftarrow M[w_j]$;
                $rel_j = \cos(\vec{w_f}, \vec{w_j})$;
                if $rel_i < rel_j$ then
                    // $w$ moves one step in the direction of $\vec{w_i} - \vec{w_j}$
                    $\vec{w_f} \leftarrow \vec{w_f} + (\vec{w_i} - \vec{w_j}) \times \Delta$;
                end if
            end for
        end for
    end for
end if
return $\vec{w_f}$;
```

3 Experiments

In this section, we present our experimental settings and results. We take the word relatedness task as the evaluation framework, and the Spearman correlation coefficient ($\rho \times 100$) is adopted to assess the quality of the learned embeddings. Also, the percentage of missed pairs (PMP) is used to evaluate the vocabulary coverage of our model. Baidu[1] search engine is used to retrieve the relevant webpages in our coarse-tuning process. All experiments use the same fine-tuning settings: $K = 50$, $\Delta = 0.1$.

We first report the performance of our algorithm in different hyper-parameters. Then, we compare the quality of the rare-word embeddings before and after the fine-tuning process to verify the effectiveness of our two-step approach. Next, we compare our algorithm with the CBOW algorithm and six state-of-the-art English rare-word embedding learning algorithms. Finally, we evaluate our algorithm on two Chinese word datasets to investigate the scalability of our approach for a language other than English.

3.1 Experimental Settings

Training corpus: We select the English Wikipedia[2] dump on April 1, 2015, as the training corpus.

Benchmark datasets: We use four benchmark datasets to perform evaluations and comparisons for different rare word representation techniques, including the Stanford Rare Word (RW) dataset (Luong et al., 2013), the Cambridge Rare Word (Card-660) dataset (Pilehvar et al., 2018), the UMNSRS dataset (Pakhomov et al., 2010) and the MayoSRS dataset (Pakhomov et al., 2011). Among them, RW (2,034 pairs) and Card-660 (660 pairs) are two general domain datasets, while UMNSRS (566 pairs) and MayoSRS (101 pairs) are two datasets in the biomedical field.

Baseline algorithms: We compare our algorithm with the CBOW algorithm (Mikolov et al., 2013a) and six rare-word learning algorithms: (i) FastText (Bojanowski et al., 2017), (ii) FCM (Schick and Schütze, 2019b), (iii) SemLand (Pilehvar and Collier., 2017), (iv) Align (Prokhorov et al., 2019), (v) ELMo (Peters et al., 2018) and (vi) BERT (Devlin et al., 2019). Among these approaches, FastText and FCM are two subword-based approaches, SemLand and Align are two lexicon-based approaches, ELMo and BERT are two pre-trained language models. The lexion WordNet (Miller, 1995) is selected as the word meaning inventory, and we use the CBOW word embedding model as the pre-trained model to induce the vectors of rare words.

3.2 Influences of Hyper-parameters

In this experiment, we investigate the influences of the two hyper-parameters in the coarse-tuning process for the quality of the learned rare-word embeddings including the number of relevant records (m) and the number of semantically related words (n), and seek the optimal range of the two hyper-

[1] http://www.baidu.com

[2] https://dumps.wikimedia.org/enwiki/20150401

parameters. We randomly select a third of records from the four benchmark datasets respectively and form four sub-datasets to evaluate the performance of our algorithm in different hyper-parameters. We first set m to 100 and change n from 1 to 10, then record the Spearman coefficients on the four sub-datasets. The dimension of all the vectors is 300, and the results are presented in Figure 2. We can see that the Spearman coefficients of our algorithm on the four sub-datasets all increase at the early stage and then decrease with the parameter n. When the parameter n is between 4 and 8, the quality of the learned rare-word embeddings is optimal. To analyze the reason, when $n < 4$, the semantic information of the related words is limited for its lower quantity, which is insufficient to predict the accurate vectors of rare words. At one extreme (when $n = 1$), the vector of a rare word directly equals to that of the only one related word without the fine-tuning process. Unless the rare word and its related word are synonymous, the rare word will obtain a wrong representation. When $n > 8$, it increases the likelihood of introducing noise words that are actually not related to the rare word into the semantic word set, which will also produce a negative effect on the right place of the rare word in the vector space.

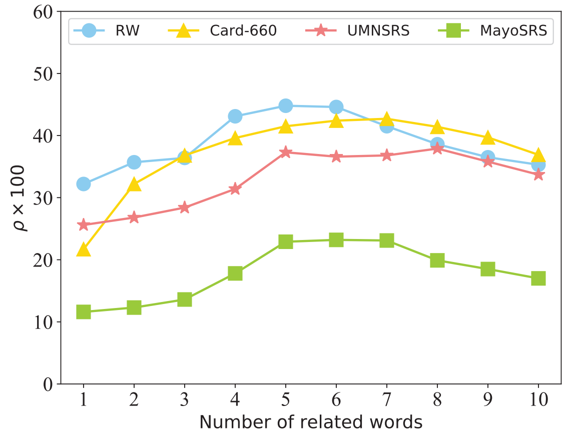

Figure 2: Spearman performance in different number of semantically related words

In the next experiment, the parameter n is set to 5, and the other parameter m is changed from 20 to 200. We record the Spearman coefficients on the four sub-datasets as well. We can see from Figure 3 that the performance reaches the peak values on the two general domain sub-datasets when the parameter m are set to 140 and 160 respectively. On the other side, the Spearman coefficients on the two biomedical field sub-datasets increase with the parameter m, then show some small fluctuations when $m > 100$. The results indicates that more retrieved records are required for general-domain rare words to obtain high-quality word representations. One reason is that the titles of the retrieved records for general-domain rare words are more likely to contain the words in the defined noise word set, and these records will be abandoned in the coarse-tuning process. Moreover, it may be unnecessary to use too many records as well because the lower-ranking records have declined in the relevance with the rare word. Based on the above results, we set m and n to 100 and 5 respectively to learn better rare-word embeddings in the follow-up experiments.

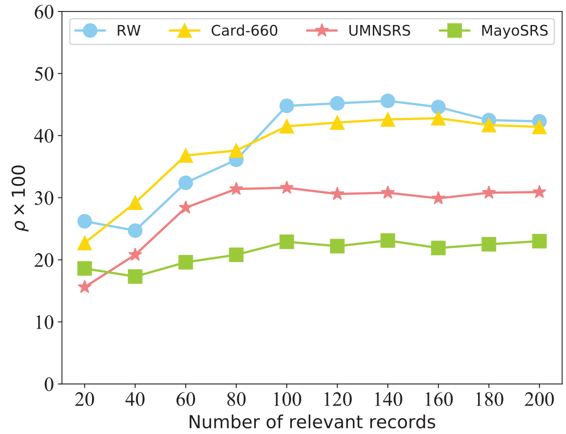

Figure 3: Spearman performance in different number of relevant records

3.3 Performance Comparison between Coarse-tuning and Fine-tuning

The coarse-tuning operation in our algorithm provides a coarse-grained vector representation from scratch for each rare word by averaging the vectors of the semantically related words, while the fine-tuning process constantly adjusts the coarse-grained embedding of the rare word to a fine-grained vector with a more suitable position in the vector space. In this experiment, we compare the Spearman correlations of the learned coarse-grained embeddings with that of the fine-grained embeddings to verify the effectiveness of the fine-tuning operation. To have a fair comparison, we report the performance of the learned rare-word embeddings on the multiple datasets in four different dimensions: 100, 200, 300 and 400.

We can see from Figure 4 that the Spearman correlations of the fine-grained embeddings outperform that of the coarse-grained embeddings on the four datasets regardless of the vector dimensions. It demonstrates that the quality of the coarse-grained embeddings can be further enhanced by the fine-tuning process with consideration of the constraints of spatial location relationships. The order information of the relevances between the rare word to its semantically related words is fully utilized to correct the vector of the rare word. Owing to the fine-tuning process, the relevance score is not required to precisely measure the relatedness between two words, but only needs to compare the relatednesses among the rare word and its semantic words relatively, which has effectively reduced the difficulty of the relevance metric design in the coarse-tuning process.

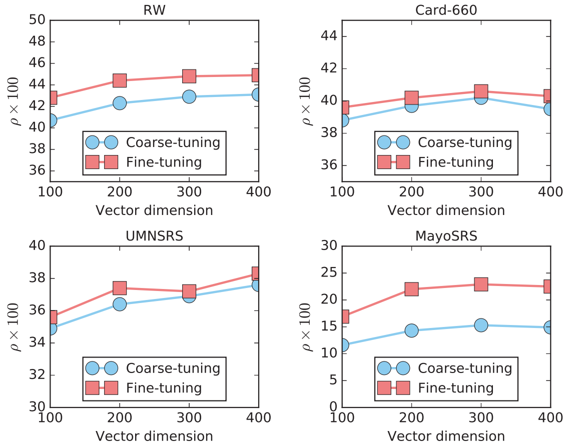

Figure 4: Performance comparison between coarse-tuning and fine-tuning over vector dimensions.

3.4 Performance Comparison with Previous Work

In this experiment, the dimension of all vectors is set to 300. Table 1 shows the comparison results, in which the best result is shown in bold. Among the baseline approaches, ELMo and BERT have the best coverage ability due to their character-based representation. Even so, we still achieve almost the same coverage performance as these two pre-trained language models except for failing to cover a tiny part (0.75%) of the word pairs in Card-660. Compared to the PMP values on domain-general datasets, the coverage performance of FastText and FCM on domain-specific datasets decline significantly. The reason is that the domain-specific terms and their subwords both rarely appear in the training corpus, which causes the vectors of many subwords in the biomedical field to be unavailable in these two models. SemLand and Align opt for WordNet as the general domain lexical resource and use the Medical Subject Headings (MeSH)[3] as the medical lexical resources. However limited by the coverage of the lexical resources, the PMP performance of these two approaches is unsteady on different datasets. We use all the information resources on the Internet to find the semantically related words of a rare word, which is far beyond the scopes of any lexical resource. Whether for domain-general rare terms or domain-specific rare terms, nearly all the vectors can be learned by our algorithm with their semantically related words.

With respect to the quality of rare word representations, our algorithm outperforms the other approaches on the four benchmark datasets. Let us go further to identify the reasons for the superiority of our algorithm. Although ELMo and BERT have the powerful coverage ability for rare words, the learned rare-word embeddings do not have high quality, especially for the domain-specific rare terms. Compared to FastText, the vector of a rare word in our algorithm is represented by its semantically related words instead of the inner subwords, which can provide more explicit semantic meanings than the subwords. FCM leverages the context information in addition to the n-gram information and learns higher-quality embeddings than FastText for the domain-general words, but fails to achieve the same performance on the domain-specific datasets. The reason is that most of the domain-specific terms are unseen in the Wikipedia corpus, so the context information is insufficient to learn the FCM word embeddings. In contrast to SemLand and Align, our algorithm is independent of specific lexical resources and has stable coverage for rare words in different fields. Therefore, our algorithm can induce eligible embeddings for more words and eventually achieve better Spearman correlations on the datasets.

3.5 Performance on Chinese rare words

To investigate the scalability of our approach for multilingual words, we evaluate our algorithm on Chinese rare words in this section. We select the Chinese Wikipedia[4] dump on November 20, 2016, as the corpus and use two Chinese benchmark datasets to perform evaluations, including the wordsim-240 (Chen et al., 2015) word

[3] https://www.nlm.nih.gov/mesh/
[4] https://dumps.wikimedia.org/zhwiki/20161120/

Table 1: The Performance of Spearman correlation and coverage on four English datasets.

Approach	RW		Card-660		UMNSRS		MayoSRS	
	PMP	$\rho \times 100$	PMP	$\rho \times 100$	PMP	$\rho \times 100$	PMP	$\rho \times 100$
CBOW	14%	35.5	54%	2.2	19%	15.9	64%	11.9
FastText	3%	38.8	5%	20.4	14%	17.6	34%	14.4
FCM	3%	39.5	5%	24.2	14%	18.8	34%	13.7
SemLand	**0%**	40.0	39%	33.2	15%	20.1	29%	10.9
Align	**0%**	42.0	39%	32.5	15%	22.4	29%	14.5
ELMo	**0%**	**44.8**	**0%**	20.2	**0%**	17.2	**0%**	7.8
BERT	**0%**	20.7	**0%**	16.2	**0%**	8.8	**0%**	7.7
Ours	**0%**	**44.8**	0.75%	**41.5**	**0%**	**37.3**	**0%**	**22.9**

pairs and the wordsim-296 (Jin and Wu, 2012). We compare our algorithm with the `CBOW` algorithm and four rare-word learning algorithms: (i) `CWE` (Chen et al., 2015), (ii) `cw2vec` (Cao et al., 2018), (iii) `ELMo` and (iv) `BERT`. The lexicon Tongyici Cilin (Tian and Zhao, 2010) is selected as the Chinese sense inventory.

We can see from Table 2 that our algorithm and the four baseline approaches all have outstanding coverage ability for the Chinese rare words. The character-based approach `CWE` and the stroke-based approach `cw2vec` fail to deduce the embedding of an outlier word "OPEC" in the wordsim-296, which brings a little loss for their performance. However, it is not a problem for our algorithm because we can extract the related words from the Internet to infer its vector including "石油 (fossil fuel)", "组织 (organization)", etc. Compared to the coverage results, our algorithm has more significant advantages over the other baseline approaches in terms of the quality and achieves the highest Spearman correlations on the two datasets. To analyze the reason, we note the extracted related words in our algorithm are more helpful to induce the embedding of a rare word than the characters in `CWE` and the strokes in `cw2vec`. Take the rare word "马拉多纳 (Maradona)" for example, it is actually not related with the character "马 (horse in Chinese)". Conversely, we can extract the semantically related words like "足球 (soccer)", "阿根廷 (Argentina)", etc, to represent this word, which can more accurately reflect its meaning, i.e., name of an athlete. Moreover, the fine-tuning operation can promote the quality of the rare-word embeddings as well.

Table 2: Evaluation results on the wordsim-240 dataset and the wordsim-296 dataset.

Approach	wordsim-240		wordsim-296	
	PMP	$\rho \times 100$	PMP	$\rho \times 100$
CBOW	4%	34.5	11%	26.1
CWE	**0%**	35.5	0.3%	38.2
cw2vec	**0%**	42.5	0.3%	43.4
ELMo	**0%**	6.0	**0%**	14.6
BERT	**0%**	15.9	**0%**	29.7
Ours	**0%**	**44.0**	**0%**	**47.2**

4 Conclusions

In this paper, we have proposed a novel algorithm to learn embeddings for rare words, which consists of a coarse-tuning process and a fine-tuning process. In the coarse-tuning process, we use an Internet search engine to retrieve webpages relevant to the rare word on Internet and extract n most related words from their titles to infer the rare word's initial vector. In the fine-tuning process, we iteratively adjust the position of the rare word in the vector space to satisfy the constraints of the spatial location relationships and get close to its semantic meaning. We evaluated our approach on multiple datasets and compared the performance with other state-of-the-art approaches. The experimental results demonstrate that our algorithm is superior to existing approaches in both the accuracy of semantic expression and the coverage for rare words.

In future, we plan to extend our algorithm to learn contextualized representations for rare words. At present, existing work on contextualized rare word representation concentrates on the improved versions of attentive mimicking (AM) architecture such as the adapted AM model (Schick and Schütze, 2020b) and the BERTRAM model (Schick and Schütze, 2020a). We consider combining our algorithm with the AM architecture, and utilize the semantically relevant information together with the surface-form information and context information to learn higher-quality context-dependent representations for rare words.

Other future work involves evaluating our algorithm leveraging other search engines (e.g., Google, Bing, etc) on multiple languages. On this basis, we seek to bring further improvements on our algorithm by selecting the most suitable search engine for a specific language to induce the embeddings of the rare words of this language.

References

D. Bahdanau, T. Bosc, S. Jastrzebski, E. Grefenstette, P. Vincent, and Y. Bengio. 2018. Learning to compute word embeddings on the fly. *arxiv:1706.00286*.

Y. Bengio, R. Ducharme, and P. Vincent. 2003. A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137–1155.

P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

S. S. Cao, W. Lu, J. Zhou, and X. L Li. 2018. cw2vec: Learning chineseword embeddings with stroke n-gram information. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI)*, New Orleans, Louisiana, USA.

X. X. Chen, Z. Y. Liu1 L. Xu, M. S. Sun, and H. B. Luan. 2015. Joint learning of character and word embeddings. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1236–1242, Buenos Aires, Argentina.

J. Devlin, M. W. Chang, K. Lee, and K. Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, Minneapolis, USA.

M. Faruqui, J. Dodge, S. K. Jauhar, C. Dyer, E. Hovy, and N. A. Smith. 2015. Retrofitting word vectors to semantic lexicons. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, pages 1606–1615, Denver, Colorado, USA.

A. Grover and J. Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 3393–3399, San Francisco, CA, USA.

T. H. Haveliwala. 2002. Topic-sensitive pagerank. In *Proceedings of 11th International Conference on World Wide Web*, pages 5053–5061, Honolulu, Hawaii, USA.

P. Jin and Y. F. Wu. 2012. Semeval-2012 task 4: Evaluating chinese word similarity. In *Proceedings of First Joint Conference on Lexical and Computational Semantics*, pages 373–374, Montreal, Canada.

M. T. Luong, R. Socher, and C. D. Manning. 2013. Better word representations with recursive neural networks for morphology. In *Proceedings of the Computational Natural Language Learning (CoNLL)*, pages 104–113, Sofia, Bulgaria.

T. Mikolov, K. Chen, G. Corrado, and J. Dean. 2013a. Efficient estimation of word representations in vector space. In *Proceedings of International Conference on Learning Representations*, Scottsdale, Arizona, USA.

T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Proceedings of International Conference on Neural Information Processing Systems (NIPS)*, pages 3111–31119, Lake Tahoe, Nevada, USA.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

S. Pakhomov, B. McInnes, T. Adam, and Y. Liu. 2010. Semantic similarity and relatedness between clinical terms: An experimental study. In *Proceedings of the Annual Symposium of the American Medical Informatics Association*, pages 572–576, Washington, D.C, USA.

S. Pakhomov, T. Pedersen, B. McInnes, G. B. Melton, A. Ruggieri, and C. G. Chute. 2011. Towards a framework for developing semantic relatedness reference standards. *Journal of Biomedical Informatics*, 4:251–265.

J. Pennington, R. Socher, and C. D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 19th Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar.

M. E. Peters, M. Neumann, M. Iyyer, and M. Gardner. 2018. Deep contextualized word representations. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, pages 2227–2237, New Orleans, Louisiana, USA.

M. T. Pilehvar and N. Collier. 2017. Inducing embeddings for rare and unseen words by leveraging lexical resources. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 388–393, Valencia, Spain.

M. T. Pilehvar, D. Kartsaklis, V. Prokhorov, and N. Collier. 2018. Card-660: Cambridge rare word dataset - a reliable benchmark for infrequentword representation models. In *Proceedings of the 23rd Conference on EmpiricalMethods in Natural Language Processing (EMNLP)*, Brussels, Belgium.

Y. Pinter, R. Guthrie, and J. Eisenstein. 2017. Mimicking word embeddings using subword rnns. In *Proceedings of the 22nd Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 102–112, Copenhagen, Denmark.

V. Prokhorov, M. T. Pilehvar, D. Kartsaklis, P. Lio, and N. Collier. 2019. Unseen word representation by aligning heterogeneous lexical semantic spaces. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI)*, pages 6900–6907, Copenhagen, Denmark.

T. Schick and H. Schütze. 2019a. Attentive mimicking: Betterword embeddings by attending to informative contexts. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, pages 489–494, Minneapolis, USA.

T. Schick and H. Schütze. 2019b. Learning semantic representations for novel words: Leveraging both form and context. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI)*, pages 6965–6973, Copenhagen, Denmark.

T. Schick and H. Schütze. 2020a. Bertram: Improved word embeddings have big impact on contextualized model performance. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3996–4007, Tokyo, Japan.

T. Schick and H. Schütze. 2020b. Rare words: A major problem for contextualized embeddings and how to fix it by attentive mimicking. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI)*, New York, NY, USA.

J. L. Tian and W. Zhao. 2010. Words similarity algorithm based on tongyici cilin in semantic web adaptive learning system. *Journal of Jilin University*, 28(6):602–608.

Z. Zhao, T. Liu, S. Li, B. F. Li, and X. Y. DU. 2017. Ngram2vec: Learning improvedword representations from ngram co-occurrence statistics. In *Proceedings of the 22nd Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 244–253, Copenhagen, Denmark.