Subword-augmented Embedding for Cloze Reading Comprehension

Zhuosheng Zhang[1,2,*], **Yafang Huang**[1,2,*], **Hai Zhao**[1,2,†], **Gongshen Liu**[3]
[1]Department of Computer Science and Engineering, Shanghai Jiao Tong University
[2]Key Laboratory of Shanghai Education Commission for Intelligent Interaction
and Cognitive Engineering, Shanghai Jiao Tong University, Shanghai, 200240, China
[3]School of Cyber Security, Shanghai Jiao Tong University, China
{zhangzs, huangyafang}@sjtu.edu.cn, zhaohai@cs.sjtu.edu.cn,
lgsheng@sjtu.edu.cn

Abstract

Representation learning is the foundation of machine reading comprehension. In state-of-the-art models, deep learning methods broadly use word and character level representations. However, character is not naturally the minimal linguistic unit. In addition, with a simple concatenation of character and word embedding, previous models actually give suboptimal solution. In this paper, we propose to use subword rather than character for word embedding enhancement. We also empirically explore different augmentation strategies on *subword-augmented embedding* to enhance the cloze-style reading comprehension model (reader). In detail, we present a reader that uses subword-level representation to augment word embedding with a short list to handle rare words effectively. A thorough examination is conducted to evaluate the comprehensive performance and generalization ability of the proposed reader. Experimental results show that the proposed approach helps the reader significantly outperform the state-of-the-art baselines on various public datasets.

1 Introduction

A recent hot challenge is to train machines to read and comprehend human languages. Towards this end, various machine reading comprehension datasets have been released, including cloze-style (Hermann et al., 2015; Hill et al., 2015; Cui et al., 2016) and user-query types (Joshi et al., 2017; Rajpurkar et al., 2016). Meanwhile, a number of deep learning models are designed to take up the challenges, most of which focus on attention mechanism (Wang et al., 2017b; Seo et al., 2017; Cui et al., 2017a; Kadlec et al., 2016; Dhingra et al., 2017; Zhang and Zhao, 2018). However, how to represent word in an effective way remains an open problem for diverse natural language processing tasks, including machine reading comprehension for different languages. Particularly, for a language like Chinese with a large set of characters (typically, thousands of), lots of which are semantically ambiguous, using either word-level or character-level embedding alone to build the word representations would not be accurate enough. This work especially focuses on a cloze-style reading comprehension task over fairy stories, which is highly challenging due to diverse semantic patterns with personified expressions and reference.

In real practice, a reading comprehension model or system which is often called *reader* in literatures easily suffers from out-of-vocabulary (OOV) word issues, especially for the cloze-style reading comprehension tasks when the ground-truth answers tend to include rare words or named entities (NE), which are hardly fully recorded in the vocabulary. This is more challenging in Chinese. There are over 13,000 characters in Chinese[1] while there are only 26 letters in English without regard to punctuation marks. If a reading comprehension system cannot effectively manage the OOV issues, the performance will not be semantically accurate for the task.

*These authors contribute equally. † Corresponding author. This paper was partially supported by National Key Research and Development Program of China (No. 2017YFB0304100), National Natural Science Foundation of China (No. 61672343 and No. 61733011), Key Project of National Society Science Foundation of China (No. 15-ZDA041), The Art and Science Interdisciplinary Funds of Shanghai Jiao Tong University (No. 14JCRZ04).

[1]Refer to the statistics of Xinhua Dictionary, version 11, published by The Commercial Press in 2014.

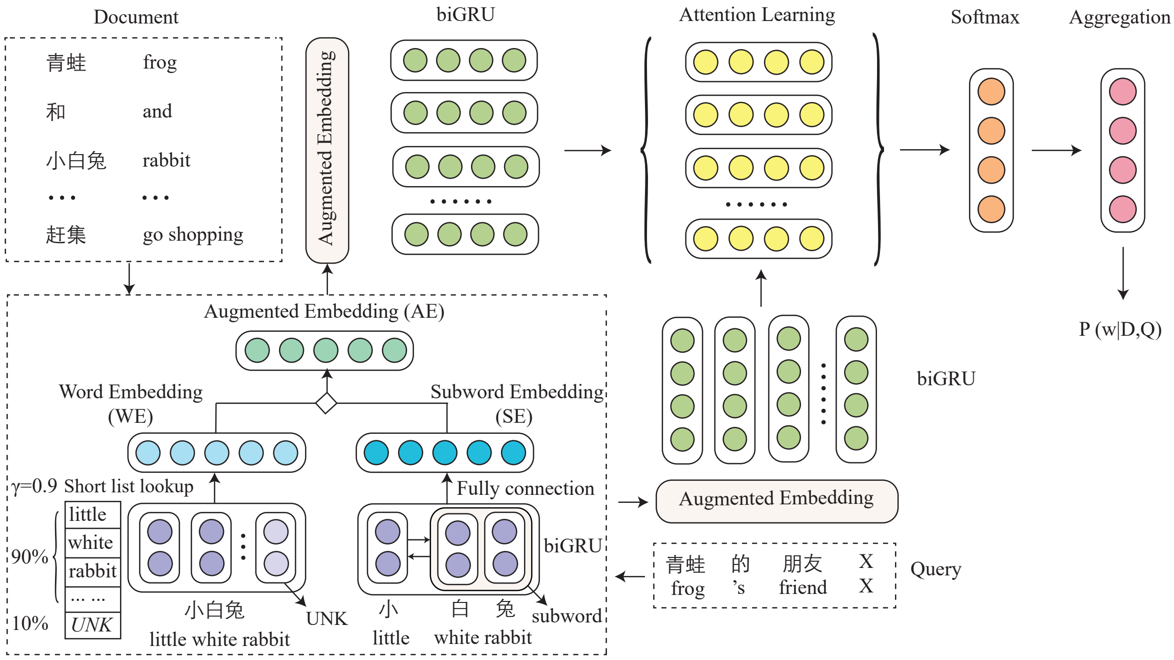


Figure 1: Architecture of the proposed Subword-augmented Embedding Reader (SAW Reader).

Commonly, words are represented as vectors using either word embedding or character embedding. For the former, each word is mapped into low dimensional dense vectors from a lookup table. Character representations are usually obtained by applying neural networks on the character sequence of the word, and their hidden states are obtained to form the representation. Intuitively, word-level representation is good at catching global context and dependency relationships between words, while character embedding helps for dealing with rare word representation.

However, the minimal meaningful unit below word usually is not character, which motivates researchers to explore the potential unit (subword) between character and word to model sub-word morphologies or lexical semantics. In fact, morphological compounding (e.g. *sunshine* or *playground*) is one of the most common and productive methods of word formation across human languages, which inspires us to represent word by meaningful sub-word units. Recently, researchers have started to work on morphologically informed word embeddings (Botha and Blunsom, 2014; Cao and Rei, 2016), aiming at better capturing syntactic, lexical and morphological information. With ready subwords, we do not have to work with characters, and segmentation could be stopped at the subword-level to reach a meaningful representation.

In this paper, we present various simple yet accurate subword-augmented embedding (SAW) strategies and propose SAW Reader as an instance. Specifically, we adopt subword information to enrich word embedding and survey different SAW operations to integrate word-level and subword-level embedding for a fine-grained representation. To ensure adequate training of OOV and low-frequency words, we employ a short list mechanism. Our evaluation will be performed on three public Chinese reading comprehension datasets and one English benchmark dataset for showing our method is also effective in multi-lingual case.

2 The Subword-augmented Word Embedding

The concerned reading comprehension task can be roughly categorized as user-query type and cloze-style according to the answer form. Answers in the former are usually a span of texts while in the cloze-style task, the answers are words or phrases which lets the latter be the harder-hit area of OOV issues, inspiring us to select the cloze-style as our testbed for SAW strategies. Our preliminary study shows even for the advanced word-character based GA reader, OOV answers still account for nearly 1/5 in the error results.

This also motivates us to explore better representations to further performance improvement.

The cloze-style task in this work can be described as a triple $< D, Q, A >$, where D is a document (context), Q is a query over the contents of D, in which a word or phrase is the right answer A. This section will introduce the proposed SAW Reader in the context of cloze-style reading comprehension. Given the triple $< D, Q, A >$, the SAW Reader will be built in the following steps.

2.1 BPE Subword Segmentation

Word in most languages usually can be split into meaningful subword units despite of the writing form. For example, "*indispensable*" could be split into the following subwords: $< in, disp, ens, able >$.

In our implementation, we adopt Byte Pair Encoding (BPE) (Gage and Philip, 1994) which is a simple data compression technique that iteratively replaces the most frequent pair of bytes in a sequence by a single, unused byte. BPE allows for the representation of an open vocabulary through a fixed-size vocabulary of variable-length character sequences, making it a very suitable word segmentation strategy for neural network models.

The generalized framework can be described as follows. Firstly, all the input sequences (strings) are tokenized into a sequence of single-character subwords, then we repeat,

1. Count all bigrams under the current segmentation status of all sequences.
2. Find the bigram with the highest frequency and merge them in all the sequences. Note the segmentation status is updating now.
3. If the merging times do not reach the specified number, go back to 1, otherwise the algorithm ends.

In (Sennrich et al., 2016), BPE is adopted to segment infrequent words into sub-word units for machine translation. However, there is a key difference between the motivations for subword segmentation. We aim to refine the word representations by using subwords, for both frequent and infrequent words, which is more generally motivated. To this end, we adaptively tokenize words in multi-granularity by controlling the merging times.

2.2 Subword-augmented Word Embedding

Our subwords are also formed as character n-grams, do not cross word boundaries. After using unsupervised segmentation methods to split each word into a subword sequence, an augmented embedding (AE) is to straightforwardly integrate word embedding $WE(w)$ and subword embedding $SE(w)$ for a given word w.

$$AE(w) = WE(w) \diamond SE(w)$$

where $\diamond$ denotes the detailed integration operation. In this work, we investigate concatenation (*concat*), element-wise summation (*sum*) and element-wise multiplication (*mul*). Thus, each document D and query Q is represented as $\mathbb{R}^{d \times k}$ matrix where d denotes the dimension of word embedding and k is the number of words in the input.

Subword embedding could be useful to refine the word embedding in a finer-grained way, we also consider improving word representation from itself. For quite a lot of words, especially those rare ones, their word embedding is extremely hard to learn due to the data sparse issue. Actually, if all the words in the dataset are used to build the vocabulary, the OOV words from the test set will not obtain adequate training. If they are initiated inappropriately, either with relatively high or low weights, they will harm the answer prediction. To alleviate the OOV issues, we keep a short list H for specific words.

$$H = \{w_1, w_2, \ldots, w_n\}$$

If w is in H, the immediate word embedding $WE(w)$ is indexed from word lookup table $M^w \in \mathbb{R}^{d \times s}$ where s denotes the size (recorded words) of lookup table. Otherwise, it will be represented as the randomly initialized default word (denoted by a specific mark UNK). Note that, this is intuitively

	CMRC-2017			PD			CFT
	Train	Valid	Test	Train	Valid	Test	human
# Query	354,295	2,000	3,000	870,710	3,000	3,000	1,953
Max # words in docs	486	481	484	618	536	634	414
Max # words in query	184	72	106	502	153	265	92
Avg # words in docs	324	321	307	379	425	410	153
Avg # words in query	27	19	23	38	38	41	20
# Vocabulary	94,352	21,821	38,704	248,160	536	634	414

Table 1: Data statistics of CMRC-2017, PD and CFT.

like "guessing" the possible unknown words (which will appear during test) from the vocabulary during training and only the word embedding of the OOV words will be replaced by UNK while their subword embedding $SE(w)$ will still be processed using the original word. In this way, the OOV words could be tuned sufficiently with expressive meaning after training. During test, the word embedding of unknown words would not severely bias its final representation. Thus, $AE(w)$ can be rewritten as

$$AE(w) = \begin{cases} WE(w) \diamond SE(w) & \text{if } w \in H \\ UNK \diamond SE(w) & \text{otherwise} \end{cases}$$

In our experiments, the short list is determined according to the word frequency. Concretely, we sort the vocabulary according to the word frequency from high to low. A frequency filter ratio γ is set to filter out the low-frequency words (rare words) from the lookup table. For example, γ=0.9 means the least frequent 10% words are replaced with the default UNK notation.

The subword embedding $SE(w)$ is generated by taking the final outputs of a bidirectional gated recurrent unit (GRU) (Cho et al., 2014) applied to the embeddings from a lookup table of subwords. The structure of GRU used in this paper are described as follows.

$$\begin{aligned} r_t &= \sigma(W_r x_t + U_r h_{t-1} + b_r), \\ z_t &= \sigma(W_z x_t + U_z h_{t-1} + b_z), \\ \tilde{h}_t &= \tanh(W_h x_t + U_h (rt \odot h_{t-1}) + b_h) \\ h_t &= (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \end{aligned}$$

where $\odot$ denotes the element-wise multiplication. r_t and z_t are the reset and update gates respectively, and $\tilde{h}_t$ are the hidden states. A bi-directional GRU (BiGRU) processes the sequence in both forward and backward directions. Subwords of each word are successively fed to forward GRU and backward GRU to obtain the internal features of two directions. The output for each input is the concatenation of the two vectors from both directions: $\overleftrightarrow{h_t} = \overrightarrow{h_t} \parallel \overleftarrow{h_t}$. Then, the output of BiGRUs is passed to a fully connected layer to obtain the final subword embedding $SE(w)$.

$$SE(w) = W \overleftrightarrow{h_t} + b$$

2.3 Attention Module

Our attention module is based on the Gated attention Reader (GA Reader) proposed by (Dhingra et al., 2017). We choose this model due to its simplicity with comparable performance so that we can focus on the effectiveness of SAW strategies. This module can be described in the following two steps. After augmented embedding, we use two BiGRUs to get contextual representations of the document and query respectively, where the representation of each word is formed by concatenating the forward and backward hidden states.

$$\begin{aligned} H_q &= \text{BiGRU}(Q) \\ H_d &= \text{BiGRU}(D) \end{aligned}$$

For each word d_i in H_d, we form a word-specific representation of the query $q_i \in H_q$ using soft attention, and then adopt element-wise product to multiply the query representation with the document word representation.

$$\alpha_i = softmax(H_q^\top d_i)$$
$$\beta_i = Q\alpha_i$$
$$x_i = d_i \odot \beta_i$$

where $\odot$ denotes the multiplication operator to model the interactions between d_i and q_i. Then, the document contextual representation $\tilde{H}_d = \{x_1, x_2, \ldots, x_k\}$ is gated by query representation.

Suppose the network has K layers. At each layer, the document representation $\tilde{H}_d$ is updated through above attention learning. After going through all the layers, our model comes to answer prediction phase. We use all the words in the document to form the candidate set C. Let q_t denote the t-th intermediate output of query representation H_q and H_D represent the full output of document representation $\tilde{H}_d$. The probability of each candidate word $w \in C$ as being the answer is predicted using a softmax layer over the inner-product between q_t and H_D.

$$p = softmax((q_t)^\top H_D)$$

where vector p denotes the probability distribution over all the words in the document. Note that each word may occur several times in the document. Thus, the probabilities of each candidate word occurring in different positions of the document are summed up for final prediction.

$$P(w|D,Q) \propto \sum_{i \in I(w,D)} p_i$$

where $I(w,d)$ denotes the set of positions that a particular word w occurs in the document D. The training objective is to maximize $\log P(A|D,Q)$ where A is the correct answer.

Finally, the candidate word with the highest probability will be chosen as the predicted answer.

$$A^* = \text{argmax}_{w \in C} P(w|D,Q)$$

Different from recent work employing complex attention mechanisms (Wang et al., 2017b; Cui et al., 2017a; Sordoni et al., 2016), our attention mechanism is much more simple with comparable performance so that we can focus on the effectiveness of SAW strategies.

3 Experiments

3.1 Dataset and Settings

To verify the effectiveness of our proposed model, we conduct multiple experiments on three Chinese Machine Reading Comprehension datasets, namely CMRC-2017 (Cui et al., 2017b), People's Daily (PD) and Children Fairy Tales (CFT) (Cui et al., 2016)[2]. In these datasets, a story containing consecutive sentences is formed as the *Document* and one of the sentences is either automatically or manually selected as the *Query* where one token is replaced by a placeholder to indicate the *answer* to fill in. Table 1 gives data statistics. Different from the current cloze-style datasets for English reading comprehension, such as CBT, Daily Mail and CNN (Hermann et al., 2015), the three Chinese datasets do not provide candidate answers. Thus, the model has to find the correct answer from the entire document.

Besides, we also use the Children's Book Test (CBT) dataset (Hill et al., 2015) to test the generalization ability in multi-lingual case. We only focus on subsets where the answer is either a common noun (CN)

[2]Note that the test set of CMRC-2017 and human evaluation test set (Test-human) of CFT are harder for the machine to answer because the questions are further processed manually and may not be accordance with the pattern of automatic questions.

Model	CMRC-2017	
	Valid	Test
Random Guess †	1.65	1.67
Top Frequency †	14.85	14.07
AS Reader †	69.75	71.23
GA Reader	72.90	74.10
SJTU BCMI-NLP †	76.15	77.73
6ESTATES PTE LTD †	75.85	74.73
Xinktech †	77.15	77.53
Ludong University †	74.75	75.07
ECNU †	77.95	77.40
WHU †	78.20	76.53
SAW Reader	**78.95**	**78.80**

Table 2: Accuracy on CMRC-2017 dataset. Results marked with † are from the latest official CMRC-2017 Leaderboard [7]. The best results are in bold face.

or NE which is more challenging since the answer is likely to be rare words. We evaluate all the models in terms of accuracy, which is the standard evaluation metric for this task.

Throughout this paper, we use the same model setting to make fair comparisons. According to our preliminary experiments, we report the results based on the following settings. The default integration strategy is *element-wise product*. Word embeddings were 200d and pre-trained by word2vec (Mikolov et al., 2013) toolkit on *Wikipedia* corpus[3]. Subword embedding were 100d and randomly initialized with the uniformed distribution in the interval [-0:05; 0:05]. Our model was implemented using the Theano[4] and Lasagne Python libraries[5]. We used stochastic gradient descent with ADAM updates for optimization (Kingma and Ba, 2014). The batch size was 64 and the initial learning rate was 0.001 which was halved every epoch after the second epoch. We also used gradient clipping with a threshold of 10 to stabilize GRU training (Pascanu et al., 2013). We use three attention layers for all experiments. The GRU hidden units for both the word and subword representation were 128. The default frequency filter proportion was 0.9 and the default merging times of BPE was 1,000. We also apply dropout between layers with a dropout rate of 0.5 [6].

3.2 Main Results

CMRC-2017 Table 2 shows our results on CMRC-2017 dataset, which shows that our SAW Reader (mul) outperforms all other single models on the test set, with 7.57% improvements compared with Attention Sum Reader (AS Reader) baseline. Although WHU's model achieves the best besides our model on the valid set with only 0.75% below ours, their result on the test set is lower than ours by 2.27%, indicating our model has a satisfactory generalization ability.

We also list different integration operations for word and subword embeddings. Table 3 shows the comparisons. From the results, we can see that Word + BPE outperforms Word + Char which indicates subword embedding works essentially. We also observe that *mul* outperforms the other two operations, *concat* and *sum*. This reveals that *mul* might be more informative than *concat* and *sum* operations. The superiority might be due to element-wise product being capable of modeling the interactions and eliminating distribution differences between word and subword embedding. Intuitively, this is also similar to endow subword-aware "attention" over the word embedding. In contrast, concatenation operation may cause too high dimension, which leads to serious over-fitting issues, and sum operation is too simple to prevent from detailed information losing.

[3] `https://dumps.wikimedia.org/`
[4] `https://github.com/Theano/Theano`
[5] `https://github.com/Lasagne/Lasagne`
[6] Our code is available at: `https://github.com/cooelf/subMrc`
[7] `http://www.hfl-tek.com/cmrc2017/leaderboard.html`

Model	Operation	CMRC-2017	
		Valid	Test
Word + Char	concat	74.80	75.13
	sum	75.40	75.53
	mul	77.80	77.93
Word + BPE	concat	75.95	76.43
	sum	76.20	75.83
	mul	**78.95**	**78.80**

Table 3: Case study on CMRC-2017.

Model	PD		CFT
	Valid	Test	Test-human
AS Reader	64.1	67.2	33.1
GA Reader	67.2	69.0	36.9
CAS Reader	65.2	68.1	35.0
SAW Reader	**72.8**	**75.1**	**43.8**

Table 4: Accuracy on PD and CFT datasets. Results of AS Reader and CAS Reader are from (Cui et al., 2016).

PD & CFT Since there is no training set for CFT dataset, our model is trained on PD training set. Note that the CFT dataset is harder for the machine to answer because the test set is further processed by human evaluation, and may not be accordance with the pattern of PD dataset. The results on PD and CFT datasets are listed in Table 4. As we see that, our SAW Reader significantly outperforms the CAS Reader in all types of testing, with improvements of 7.0% on PD and 8.8% on CFT test sets, respectively. Although the domain and topic of PD and CFT datasets are quite different, the results indicate that our model also works effectively for out-of-domain learning.

CBT To verify if our method can only work for Chinese, we also evaluate the effectiveness of the proposed method on benchmark English dataset. We use CBT dataset as our testbed to evaluate the performance. For a fair comparison, we simply set the same parameters as before. Table 5 shows the results. We observe that our model outperforms most of the previously public works, with 2.4 % gains on the CBT-NE test set compared with GA Reader which adopts word and character embedding concatenation. Our SAW Reader also achieves comparable performance with FG Reader who adopts neural gates to combine word-level and character-level representations with assistance of extra features including NE, POS and word frequency while our model is much simpler and faster. This result shows our SAW Reader is not restricted to Chinese reading comprehension, but also for other languages.

4 Analysis

4.1 Merging Times of BPE

The vocabulary size could seriously involve the segmentation granularity. For BPE segmentation, the resulted subword vocabulary size is equal to the merging times plus the number of single-character types. To have an insight of the influence, we adopt merge times from 0 to $20k$, and conduct quantitative study on CMRC-2017 for BPE segmentation. Figure 2 shows the results. We observe that when the vocabulary size is $1k$, the models could obtain the best performance. The results indicate that for a task like reading comprehension the subwords, being a highly flexible grained representation between character and word, tends to be more like characters instead of words. However, when the subwords completely fall into characters, the model performs the worst. This indicates that the balance between word and character is quite critical and an appropriate grain of character-word segmentation could essentially improve the word representation.

Model	CBT-NE		CBT-CN	
	Valid	Test	Valid	Test
Human ‡	-	81.6	-	81.6
LSTMs ‡	51.2	41.8	62.6	56.0
MemNets ‡	70.4	66.6	64.2	63.0
AS Reader ‡	73.8	68.6	68.8	63.4
Iterative Attentive Reader ‡	75.2	68.2	72.1	69.2
EpiReader ‡	75.3	69.7	71.5	67.4
AoA Reader ‡	77.8	72.0	72.2	69.4
NSE ‡	78.2	73.2	74.3	71.9
FG Reader ‡	**79.1**	**75.0**	**75.3**	**72.0**
GA Reader ‡	76.8	72.5	73.1	69.6
SAW Reader	78.5	74.9	75.0	71.6

Table 5: Accuracy on CBT dataset. Results marked with ‡ are of previously published works (Dhingra et al., 2017; Cui et al., 2016; Yang et al., 2017).

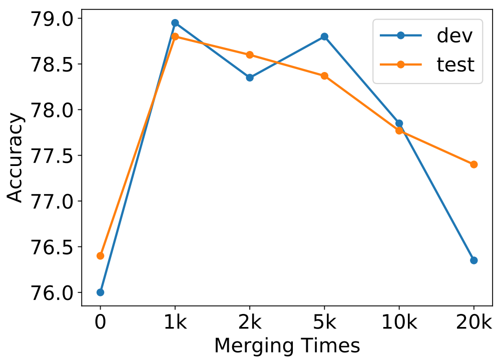


Figure 2: Case study of the subword vocabulary size of BPE.

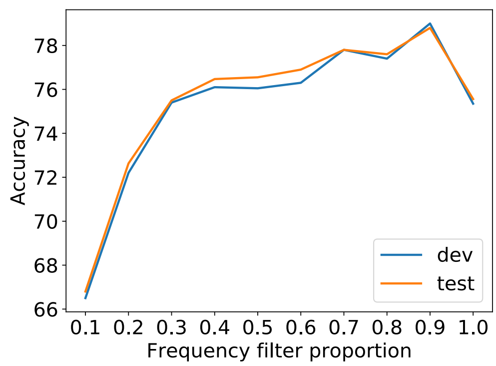


Figure 3: Quantitative study on the influence of the short list.

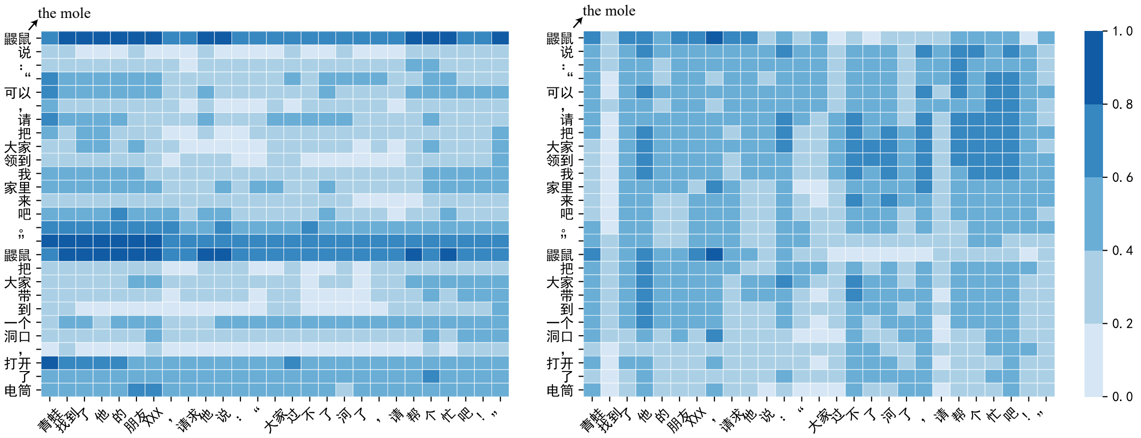
4.2 Filter Mechanism

To investigate the impact of the short list to the model performance, we conduct quantitative study on the filter ratio from [0.1, 0.2, . . . , 1]. The results on the CMRC-2017 dataset are depicted in Figure 3. As we can see that when $\gamma = 0.9$ our SAW reader can obtain the best performance, showing that building the vocabulary among all the training set is not optimal and properly reducing the frequency filter ratio can boost the accuracy. This is partially attributed to training the model from the full vocabulary would cause serious over-fitting as the rare words representations can not obtain sufficient tuning. If the rare words are not initialized properly, they would also bias the whole word representations. Thus a model without OOV mechanism will fail to precisely represent those inevitable OOV words from test sets.

4.3 Subword-Augmented Representations

In text understanding tasks, if the ground-truth answer is OOV word or contains OOV word(s), the performance of deep neural networks would severely drop due to the incomplete representation, especially for cloze-style reading comprehension task where the answer is only one word or phrase. In CMRC-2017, we observe questions with OOV answers (denoted as "OOV questions") account for 17.22% in the error results of the best Word + Char embedding based model. With BPE subword embedding, 12.17% of these "OOV questions" could be correctly answered. This shows the subword representations could be essentially useful for modeling rare and unseen words.

To analyze the reading process of SAW Reader, we draw the attention distributions at intermediate



(a) Embedding of Document and query

(b) Final document and query representation

Doc (extract): The mole said, "That's fine, please bring them to my house." The mole took everyone to a hole, turned on the flashlight and asked the little white rabbit, the hedgehog, the big ant and the frog to follow him, saying, "Don't be afraid, just go ahead."
Query: The frog found his friend ______ and told him, We cannot get across the river. Please give us a hand!

Figure 4: Pair-wise attention visualization.

layers as shown in Figure 4. We observe the salient candidates in the document can be focused after the pair-wise matching of document and query and the right answer ("*The mole*") could obtain a high weight at the very beginning. After attention learning, the key evidence of the answer would be collected and irrelevant parts would be ignored. This shows our SAW Reader is effective at selecting the vital points at the fundamental embedding layer, guiding the attention layers to collect more relevant pieces.

5 Related Work

5.1 Machine Reading Comprehension

Recently, many deep learning models have been proposed for reading comprehension (Sordoni et al., 2016; Trischler et al., 2016; Wang and Jiang, 2016; Munkhdalai and Yu, 2017; Wang et al., 2017a; Dhingra et al., 2017; Zhang et al., 2018b; Wang et al., 2018b). Notably, Chen et al. (2016) conducted an in-depth and thoughtful examination on the comprehension task based on an attentive neural network and an entity-centric classifier with a careful analysis based on handful features. Kadlec et al. (2016) proposed the Attention Sum Reader (AS Reader) that uses attention to directly pick the answer from the context, which is motivated by the Pointer Network (Vinyals et al., 2015). Instead of summing the attention of query-to-document, GA Reader (Dhingra et al., 2017) defined an element-wise product to endowing attention on each word of the document using the entire query representation to build query-specific representations of words in the document for accurate answer selection. Wang et al. (2017b) employed gated self-matching networks (R-net) on passage against passage itself to refine passage representation with information from the whole passage. Cui et al. (2017a) introduced an "attended attention" mechanism (AoA) where query-to-document and document-to-query are mutually attentive and interactive to each other.

5.2 Augmented Word Embedding

Distributed word representation plays a fundamental role in neural models (Cai and Zhao, 2016; Qin et al., 2016; Zhao et al., 2017; Peters et al., 2018; He et al., 2018; Wang et al., 2018a; Bai and Zhao, 2018; Zhang et al., 2018a). Recently, character embeddings are widely used to enrich word representations (Kim et al., 2016; Yang et al., 2017; Luong and Manning, 2016; Huang et al., 2018). Yang et al. (2017) explored a fine-grained gating mechanism (FG Reader) to dynamically combine word-level and character-level representations based on properties of the words. However, this method is computationally complex and it is not end-to-end, requiring extra labels such as NE and POS tags. Seo et al. (2017)

concatenated the character and word embedding to feed a two-layer Highway Network.

Not only for machine reading comprehension tasks, character embedding has also benefit other natural language process tasks, such as word segmentation (Cai et al., 2017), machine translation (Luong and Manning, 2016), tagging (Yang et al., 2016; Li et al., 2018) and language modeling (Verwimp et al., 2017; Miyamoto and Cho, 2016). However, character embedding only shows marginal improvement due to a lack internal semantics. Lexical, syntactic and morphological information are also considered to improve word representation (Cao and Rei, 2016; Bergmanis and Goldwater, 2017). Bojanowski et al. (2017) proposed to learn representations for character n-gram vectors and represent words as the sum of the n-gram vectors. Avraham and Goldberg (2017) built a model inspired by (Joulin et al., 2017), who used morphological tags instead of n-grams. They jointly trained their morphological and semantic embeddings, implicitly assuming that morphological and semantic information should live in the same space. However, the linguistic knowledge resulting subwords, typically, morphological suffix, prefix or stem, may not be suitable for different kinds of languages and tasks. Sennrich et al. (2016) introduced the byte pair encoding (BPE) compression algorithm into neural machine translation for being capable of open-vocabulary translation by encoding rare and unknown words as subword units. Instead, we consider refining the word representations for both frequent and infrequent words from a computational perspective. Our proposed subword-augmented embedding approach is more general, which can be adopted to enhance the representation for each word by adaptively altering the segmentation granularity in multiple NLP tasks.

6 Conclusion

This paper presents an effective neural architecture, called subword-augmented word embedding to enhance the model performance for the cloze-style reading comprehension task. The proposed SAW Reader uses subword embedding to enhance the word representation and limit the word frequency spectrum to train rare words efficiently. With the help of the short list, the model size will also be reduced together with training speedup. Unlike most existing works, which introduce either complex attentive architectures or many manual features, our model is much more simple yet effective. Giving state-of-the-art performance on multiple benchmarks, the proposed reader has been proved effective for learning joint representation at both word and subword level and alleviating OOV difficulties.

References

Oded Avraham and Yoav Goldberg. 2017. The interplay of semantics and morphology in word embeddings. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2017)*, pages 422–426.

Hongxiao Bai and Hai Zhao. 2018. Deep enhanced representation for implicit discourse relation recognition. In *Proceedings of the 27th International Conference on Computational Linguistics (COLING 2018)*.

Toms Bergmanis and Sharon Goldwater. 2017. From segmentation to analyses: a probabilistic model for unsupervised morphology induction. In *Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 337–346.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistic (TACL)*, 5:135–146.

Jan A. Botha and Phil Blunsom. 2014. Compositional morphology for word representations and language modelling. *Proceedings of the 31st International Conference on Machine Learning (ICML 2014)*, 32:1899–1907.

Deng Cai and Hai Zhao. 2016. Neural word segmentation learning for Chinese. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*, pages 409–420.

Deng Cai, Hai Zhao, Zhisong Zhang, Yuan Xin, Yongjian Wu, and Feiyue Huang. 2017. Fast and accurate neural word segmentation for Chinese. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017)*, pages 608–615.

Kris Cao and Marek Rei. 2016. A joint model for word embedding and word morphology. In *The Workshop on Representation Learning for NLP*, pages 18–26.

Danqi Chen, Jason Bolton, and Christopher D Manning. 2016. A thorough examination of the CNN/Daily Mail reading comprehension task. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*, pages 2358–2367.

Kyunghyun Cho, Bart Van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, pages 1724–1734.

Yiming Cui, Ting Liu, Zhipeng Chen, Shijin Wang, and Guoping Hu. 2016. Consensus attention-based neural networks for Chinese reading comprehension. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers (COLING 2016)*, pages 1777–1786.

Yiming Cui, Zhipeng Chen, Si Wei, Shijin Wang, Ting Liu, and Guoping Hu. 2017a. Attention-over-attention neural networks for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017)*, pages 1832–1846.

Yiming Cui, Ting Liu, Zhipeng Chen, Wentao Ma, Shijin Wang, and Guoping Hu. 2017b. Dataset for the first evaluation on Chinese machine reading comprehension. *arXiv preprint arXiv:1511.02301*.

Bhuwan Dhingra, Hanxiao Liu, Zhilin Yang, William W. Cohen, and Ruslan Salakhutdinov. 2017. Gated-attention readers for text comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017)*, pages 1832–1846.

Gage and Philip. 1994. A new algorithm for data compression. *C Users Journal*, 12(2):23–38.

Shexia He, Zuchao Li, Hai Zhao, Hongxiao Bai, and Gongshen Liu. 2018. Syntax for semantic role labeling, to be, or not to be. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL 2018)*.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems 28 (NIPS 2015)*, pages 1693–1701.

Felix Hill, Antoine Bordes, Sumit Chopra, and Jason Weston. 2015. The goldilocks principle: Reading children's books with explicit memory representations. *arXiv preprint arXiv:1511.02301*.

Yafang Huang, Zuchao Li, Zhuosheng Zhang, and Hai Zhao. 2018. Moon IME: neural-based chinese pinyin aided input method with customizable association. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL 2018), System Demonstration*.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017)*, pages 1601–1611.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2017)*, pages 427–431.

Rudolf Kadlec, Martin Schmid, Ondrej Bajgar, and Jan Kleindienst. 2016. Text understanding with the attention sum reader network. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*, pages 908–918.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M Rush. 2016. Character-aware neural language models. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI 2016)*, pages 2741–2749.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Haonan Li, Zhisong Zhang, Yuqi Ju, and Hai Zhao. 2018. Neural character-level dependency parsing for Chinese. In *The Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)*.

Minh-Thang Luong and Christopher D Manning. 2016. Achieving open vocabulary neural machine translation with hybrid word-character models. *arXiv preprint arXiv:1604.00788*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Yasumasa Miyamoto and Kyunghyun Cho. 2016. Gated word-character recurrent language model. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP 2016)*, pages 1992–1997.

Tsendsuren Munkhdalai and Hong Yu. 2017. Reasoning with memory augmented neural networks for language comprehension. *Proceedings of the International Conference on Learning Representations (ICLR 2017).*

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks. In *International Conference on International Conference on Machine Learning*, volume 28, page 13101318.

M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365.*

Lianhui Qin, Zhisong Zhang, and Hai Zhao. 2016. A stacking gated neural architecture for implicit discourse relation classification. In *Conference on Empirical Methods in Natural Language Processing (EMNLP 2016)*, pages 2263–2270.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP 2016)*, pages 2383–2392.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*, pages 1715–1725.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Bidirectional attention flow for machine comprehension. In *Proceedings of the International Conference on Learning Representations (ICLR 2017).*

Alessandro Sordoni, Philip Bachman, Adam Trischler, and Yoshua Bengio. 2016. Iterative alternating neural attention for machine reading. *arXiv preprint arXiv:1606.02245.*

Adam Trischler, Zheng Ye, Xingdi Yuan, and Kaheer Suleman. 2016. Natural language comprehension with the epireader. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP 2016)*, pages 128–137.

Lyan Verwimp, Joris Pelemans, Hugo Van Hamme, and Patrick Wambacq. 2017. Character-word LSTM language models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2017)*, pages 417–427.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. *In Advances in Neural Information Processing Systems*, pages 2692–2700.

Shuohang Wang and Jing Jiang. 2016. Machine comprehension using Match-LSTM and answer pointer. *Proceedings of the International Conference on Learning Representations (ICLR 2016).*

Bingning Wang, Kang Liu, and Jun Zhao. 2017a. Conditional generative adversarial networks for commonsense machine comprehension. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI 2017)*, pages 4123–4129.

Wenhui Wang, Nan Yang, Furu Wei, Baobao Chang, and Ming Zhou. 2017b. Gated self-matching networks for reading comprehension and question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017)*, pages 189–198.

Rui Wang, Hai Zhao, Sabine Ploux, Bao-Liang Lu, Masao Utiyama, and Eiichiro Sumita. 2018a. Graph-based bilingual word embedding for statistical machine translation. *ACM Transaciton on Asian and Low-Resource Language Information Processing*, 17(4).

Yizhong Wang, Kai Liu, Jing Liu, Wei He, Yajuan Lyu, Hua Wu, Sujian Li, and Haifeng Wang. 2018b. Multi-passage machine reading comprehension with cross-passage answer verification.

Zhilin Yang, Ruslan Salakhutdinov, and William Cohen. 2016. Multi-task cross-lingual sequence tagging from scratch. *arXiv preprint arXiv:1603.06270.*

Zhilin Yang, Bhuwan Dhingra, Ye Yuan, Junjie Hu, William W. Cohen, and Ruslan Salakhutdinov. 2017. Words or characters? fine-grained gating for reading comprehension. In *Proceedings of the International Conference on Learning Representations (ICLR 2017).*

Zhuosheng Zhang and Hai Zhao. 2018. One-shot learning for question-answering in gaokao history challenge. In *Proceedings of the 27th International Conference on Computational Linguistics (COLING 2018)*.

Zhuosheng Zhang, Jiangtong Li, Hai Zhao, and Bingjie Tang. 2018a. Sjtu-nlp at semeval-2018 task 9: Neural hypernym discovery with term embeddings. In *Proceedings of the 12th International Workshop on Semantic Evaluation (SemEval 2018), Workshop of NAACL-HLT 2018*.

Zhuosheng Zhang, Jiangtong Li, Pengfei Zhu, and Hai Zhao. 2018b. Modeling multi-turn conversation with deep utterance aggregation. In *Proceedings of the 27th International Conference on Computational Linguistics (COLING 2018)*.

Hai Zhao, Deng Cai, Yang Xin, Yuzhu Wang, and Zhongye Jia. 2017. A hybrid model for Chinese spelling check. *ACM Transactions on Asian Low-Resource Language Information Process*, pages 1–22.