Effective Subword Segmentation for Text Comprehension

Zhuosheng Zhang, Hai Zhao*, Kangwei Ling, Jiangtong Li, Zuchao Li, Shexia He
Department of Computer Science and Engineering, Shanghai Jiao Tong University
Key Laboratory of Shanghai Education Commission for Intelligent Interaction
and Cognitive Engineering, Shanghai Jiao Tong University, Shanghai, 200240, China
zhangzs@sjtu.edu.cn, zhaohai@cs.sjtu.edu.cn,
{kevinling, keep_moving-lee, charlee, heshexia}@sjtu.edu.cn

Abstract
Character-level representations have been broadly adopted to alleviate the problem of effectively representing rare or complex words. However, character itself is not a natural minimal linguistic unit for representation or word embedding composing due to ignoring the linguistic coherence of consecutive characters inside word. This paper presents a general subword-augmented embedding framework for learning and composing computationally-derived subword-level representations. We survey a series of unsupervised segmentation methods for subword acquisition and different subword-augmented strategies for text understanding, showing that subword-augmented embedding significantly improves our baselines in multiple text understanding tasks on both English and Chinese languages.

Introduction
The fundamental part of deep learning methods applied to natural language processing (NLP), distributed word representation, namely, word embedding, provides a basic solution to text representation for NLP tasks and have proven useful in various applications, including text entailment [Cheng, Dong, and Lapata 2016], Wang, Hamza, and Florian 2017] and reading comprehension [Yang et al. 2017], Seo et al. 2017], Wang et al. 2017], Clark and Gardner 2018], Zhang and Zhao 2018], Tan et al. 2018]. However, deep learning based NLP models usually suffer from rare and out-of-vocabulary (OOV) word representation. Besides, most word embedding approaches treat word forms as atomic units, which is spoiled by many words that actually have a complex internal structure. Especially, rare words like morphologically complex words and named entities, are often expressed poorly due to data sparsity. Actually, plenty of words share some conjunct written units, such as morphemes, stems and affixes. The models would benefit a lot from distilling these salient units automatically.

Character-level embedding has been broadly used to refine the word representation [Yang et al. 2017], Kim et al. 2016, Luong and Manning 2016, Li et al. 2018], showing beneficially complementary to word representations. Concretely, each word is split into a sequence of characters. Character representations are obtained by applying neural networks on the character sequence of the word, and their hidden states form the representation.

However, character is not the natural minimum linguistic unit, which makes it quite valuable to explore the potential unit (subword) between character and word to model sub-word morphologies or lexical semantics. For English, there are only 26 letters. Using such a small character vocabulary to form the word representations could be too insufficient and coarse. Even for a language like Chinese with a large set of characters (typically, thousands of), lots of which are semantically ambiguous, using character embedding below the word-level to build the word representations would not be accurate enough, either. For example, for an internet neologism 老司机 (experienced driver), the characters <老(experienced, old)> , 司/manage, 机(machine)> would be somewhat from the meaning of the word while the subwords <老(experienced, old)> , 司/manager, 机/vehicle> with proper syntactic and semantic decomposition give exactly the minimal meaningful units below the word-level which surely improve the later word representation. Thus, in either type of languages, effective representation cannot be done accurately only via the character based process.

In fact, morphological compounding (e.g. sunshine or playground) is one of the most common and productive methods of word formation across human languages, and most of rare or OOV words can be segmented into meaningful fine-grained subword units for accurate learning and representation, which inspires us to represent word by meaningful sub-word units. Recently, researchers have started to work on morphologically informed word representations [Hammarstrom and Borin 2011], Botha and Blunsom 2014], Cao and Rei 2016], Bergmanis and Goldwater 2017], aiming at better capturing syntactic, lexical and morphological information. With flexible subwords from either source, we do not necessarily need to work with characters, and segmentation could be stopped at the subword-level. With related characters grouping into subword, we hopefully reach a meaningful minimal representation unit.

Splitting a word into sub-word level subwords and using these subwords to augment the word representation may recover the lost syntactic or semantic information that is supposed to be delivered by subwords. For example, understanding could be split into the following subwords: <under, stand, ing>. Previous work usually considered prior linguis-
tic knowledge based methods to tokenize each word into subwords (namely, morphological based subword). However, such treatment may encounter two main inconveniences. First, the linguistic knowledge resulting subwords, typically, morphological suffix, prefix or stem, may not be suitable for the targeted NLP tasks. Second, linguistic knowledge or related annotated lexicons or corpora even may not be available for a specific language or task. Thus in this work we consider computationally motivated subword tokenization approaches instead.

We present a unified representation learning framework to sub-word level information enhanced text understanding and surveys various computationally motivated segmentation methods. Therefore, we consider the subword as the basic unit in our models and manipulate the neural architecture accordingly. The proposed subword-augmented embedding will be evaluated on text understanding tasks, including text entailment and reading comprehension, both of which are quite challenging due to the need of accurate lexical-level representation. We empirically survey various subword segmentation methods from a computational perspective and investigate the better way to enhance the tasks with thoughtful analysis and case studies.

Related Work

Distributed word representation plays a fundamental role in neural models (Zhang et al. 2018a, Wang et al. 2018, Zhang et al. 2018b). Previous work has shown word representations in NLP tasks can benefit from character-level models, which aim at learning language representations directly from characters. Character-level features have been widely used in language modeling (Miyamoto and Cho 2016, Verwimp et al. 2017), tagging (Yang, Salakhutdinov, and Cohen 2016, Ling et al. 2015), machine translation (Luong and Manning 2016, Sennrich, Haddow, and Birch 2016) and reading comprehension (Yang et al. 2017, Seo et al. 2017). Seo et al. (2017) concatenated the character and word embedding to feed a two-layer Highway Network. Cai et al. (2017) presented a greedy neural word segmenter to balance word and character embeddings. High-frequency word embeddings are attached to character embedding via average pooling while low-frequency words are represented as character embedding. Miyamoto and Cho (2016) introduced a recurrent neural network language model with LSTM units and a word-character gate to adaptively find the optimal mixture of the character-level and word-level inputs. Yang et al. (2017) explored a fine-grained gating mechanism to dynamically combine word-level and character-level representations based on properties of the words (e.g. named entity and part-of-speech tags).

However, character embeddings only show marginal improvement due to a lack internal semantics. Recently, many techniques were proposed to enrich word representations with sub-word information. Bojanowski et al. (2017) proposed to learn representations for character $n$-gram vectors and represent words as the sum of the $n$-gram vectors. Avra-

\[ w_{t} = \text{arg max}_{w \in W} g(w) \]  

\[ \{w_{t}, t_{t}\} = \text{arg max}_{w \in W} g(w) \]  

This is a greedy algorithm with respect to a goodness score. It works on $T$ to output the best current subword $w_{t}$ repeatedly with $T = t_{t}$ for the next round as follows,

\[ \{w_{t}, t_{t}\} = \text{arg max}_{w \in W} g(w) \]  

with each $\{w, g(w)\} \in W$.

In this work, we additionally introduce the third segmentation algorithm.

Unsupervised Subword Segmentation

To segment subwords from word that is regarded as character sequence, we adopt and extend the generalized unsupervised segmentation framework proposed by Zhao and Kit (2008), which was originally designed only for Chinese word segmentation.

The generalized framework can be divided into two collocative parts, goodness measurement which evaluates how likely a subword is to be a ‘proper’ one, and a segmentation or decoding algorithm. The framework generally works in two steps. First, a goodness score $g(w_{t})$ is computed for each $n$-gram $w_{t}$ (in this paper gram always refers to character) using the selected goodness measure to form a dictionary $W = \{w_{t}, g(w_{t})\}_{t=1,...,n}$. Then segmentation or decoding method is applied to tokenize words into subwords based on the dictionary.

Viterbi

This style of segmentation is to search for a segmentation with the largest goodness score sum for an input unsegmented sequence $T$ (to be either words or Chinese sentence).

Maximal-Matching (MM)

This is a greedy algorithm with respect to a goodness score. It works on $T$ to output the best current subword $w_{t}$ repeatedly with $T = t_{t}$ for the next round as follows,
Byte Pair Encoding (BPE)  Byte Pair Encoding (BPE) (Gage and Philip 1994) is a simple data compression technique that iteratively replaces the most frequent pair of bytes in a sequence by a single, unused byte. Different from the previous two algorithms that segment the input sequence into pieces in a top-down way, BPE segmentation actually merges a full single-character segmentation to a reasonable segmentation in a bottom-up way. We formalize the generalized BPE style segmentation in the following.

At the very beginning, all the input sequences are tokenized into a sequence of single-character subwords, then we repeat,

1. Calculate the goodness scores of all bigrams under the current segmentation status of all sequences.
2. Find the bigram with the highest goodness score and merge them in all the sequences. Note the segmentation status has been updated at this time.

3. If the merging times does not reach the specified number, go back to 1, otherwise the algorithm ends.

In our work, we investigate three types of goodness measures to evaluate subword likelihood, namely Frequency, Accessor Variety and Description Length Gain.

Frequency (FRQ)  FRQ is simply defined as the counting in the entire corpus for each n-gram being subword candidate. We take a logarithmic form as the goodness score,

\[ g_{\text{FRQ}}(w) = \log(p(w)) \]  

(2)

where \( p(w) \) is \( w \)'s frequency in the corpus.

Accessor Variety (AV)  AV is proposed by (Feng et al. 2004) to measure how likely a subword is a true word. The AV of a subword \( x_{i..j} \) is defined as

\[ \text{AV}(x_{i..j}) = \min \{ L_{\text{av}}(x_{i..j}), R_{\text{av}}(x_{i..j}) \} \]  

(3)

where the left and right accessor variety \( L_{\text{av}}(x_{i..j}) \) and \( R_{\text{av}}(x_{i..j}) \) are, respectively, the number of distinct predecessor and successor characters. The same as FRQ, the goodness score is taken in logarithmic form, \( g_{\text{AV}}(w) = \log(\text{AV}(w)) \).

Description Length Gain (DLG)  Wilks 1999 proposed this goodness measure for compression-based segmentation. The DLG replaces all occurrences of \( x_{i..j} \) from a corpus \( X = x_1x_2...x_n \) as a subword and is computed by

\[ \text{DLG}(x_{i..j}) = L(X) - L(X[r \rightarrow x_{i..j}] \oplus x_{i..j}) \]  

(4)

where \( X[r \rightarrow x_{i..j}] \) represents the resultant corpus by replacing all items of \( x_{i..j} \) with a new symbol \( r \) throughout \( X \) and \( \oplus \) denotes the concatenation. \( L(\cdot) \) is the empirical description length of a corpus in bits that can be estimated by

the Shannon–Fano code or Huffman code, following classic information theory (Shannon 1948).

\[ L(X) = -\sum_{x \in V} p(x) \log_2 p(x) \]  

(5)

where \(|\cdot|\) denotes the string length, \( V \) is the vocabulary of \( X \) and \( \hat{p}(x) \) is \( x \)'s frequency in \( X \). The goodness score is given by \( g_{\text{DLG}}(w) = \text{DLG}(w) \).

It is easy to find that BPE style segmentation with FRQ (denoted as BPE-FRQ) as the goodness score will result in the BPE subword encoding in (Sennrich, Haddow, and Birch 2016) for infrequent (rare or OOV) word representation in neural machine translation. Instead, 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.

Subword-augmented Embedding

Our subwords are also formed as character n-gram, do not cross word boundaries. After splitting each word into a subword sequence, an augmented embedding (AE) is formed to straightforwardly integrate word embedding \( W_E(w) \) and subword embedding \( SE(w) \) for a given word \( w \).

\[ AE(w) = W_E(w) \odot SE(w) \]

where \( \odot \) denotes the integration strategy. In this work, we investigate concatenation (concat), element-wise summation (sum) and element-wise multiplication (mul). Thus, each sentence \( D \) 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 sequence \( w = \{x_1, x_2, \ldots, x_t\} \) is embedded into vectors \( M \) using a lookup table, which is taken as the inputs to the CNN, and whose size is the input channel size of the CNN. Let \( W_j \) denote the Filter matrices of width \( l \), the substring vectors will be transformed to sequences \( c_j \left( j \in [1, l] \right) \):

\[ c_j = [\ldots; \text{tanh}(W_j \cdot M[i+i+l-1] + b_j); \ldots] \]

where \( [i : i + l - 1] \) indexes the convolution window. A one-max-pooling operation is adopted after convolution \( s_j = \max(c_j) \). The outputs of \( k \) filters are concatenated,

\[ u = [s_1 \oplus \cdots \oplus s_j \oplus \cdots \oplus s_l] \]

to feed to a highway network (Srivastava, Greff, and Schmidhuber 2015)

\[ g = \sigma(W_gu^T + b_g) \]

\[ SE = g \odot \text{Sigmoid}(W_ku^T + b_k) + (1 - g) \odot u \]

Experiments

In this section, we evaluate the performance of subword-augmented embedding on two kinds of challenging text understanding tasks, text entailment and reading comprehension. Both of the concerned tasks are quite challenging, let alone the latest performance improvement has been already very marginal. However, we present a new solution in a new
tasks. The default integration strategy is concatenation for 1 entailment task and 1 embedding method for text understanding. Can using subword-augmented embedding enhance the concerned tasks? Can using subword embedding help effectively model OOV or rare words? Which is the best unsupervised subword segmentation method for text understanding? Which is the best strategy to integrate word and subword embedding?

The default subword vocabulary size is set $10^k$ for text entailment task and $1k$ for the two reading comprehension tasks. The default integration strategy is concatenation for the following experiments. The above choices are based on the model performance on the development set and the detailed analysis will be given in Section 6. Word embeddings are 200d and pre-trained by word2vec [Mikolov et al. 2013] toolkit on Wikipedia corpus. Both character and subword embeddings are also 200d and randomly initialized with the uniformed distribution in the interval [-0.05; 0.05]. Note that character could be regarded as the minimal case of subwords, we separately depict them in our experiments for better comparison and convenient demonstration.

In our preliminary experiments, we thoroughly explore all nine subword segmentation methods by considering there are three segmentation algorithms and three goodness measures. We find that all Viterbi based segmentation fails to show satisfactory performance, and we only report three best performing segmentation-goodness collocations for each task. Our baseline models are selected due to their simplicity and state-of-the-art performance in each task. We are interested in a subword-based framework that performs robustly across a diverse set of tasks. To this end, we follow the same hyper-parameters or each baseline model as the original settings from their corresponding literatures except those specified (e.g. subword dimension, integration strategy). Since ensemble systems are commonly integrated with multiple heterogeneous models and resources and thus not completely comparable, we only focus on the evaluations on single models.

### Text Entailment

Textual entailment is the task of determining whether a hypothesis is entailment, contradiction and neutral, given a premise. The Stanford Natural Language Inference (SNLI) corpus [Bowman et al. 2015] provides approximately 570k hypothesis/premise pairs.

Our baseline model is Enhanced Sequential Inference Model (ESIM) [Chen et al. 2017] which employs a biLSTM to encode the premise and hypothesis, followed by an attention layer, a local inference layer, an inference composition layer. To keep the model simplicity and concentrate on the performance of subword units, we do not integrate extra syntactic parsing features or increase the dimension of word embeddings. However, with the subword augmentation, the simple sequential encoding model yields substantial gains and achieves competitive performance with more complex state-of-the-art models.

The dimensions for all the LSTM and fully connection layers were 300. We set the dropout rate to 0.5 for each LSTM layer and the fully connected layers. All feed forward layers used ReLU activations. Parameters were optimized using Adam [Kingma and Ba 2014] with gradient norms clipped at 5.0. The initial learning rate was 0.001, which was halved every epoch after the second epoch. The batch size was 32.

Results in Table 1 show that, subword-augmented embedding can boost our baseline (Word + Char) by +0.95% on the test set. Among the subword algorithms, BPE-DLG performs the best whose key difference with other approaches is BPE-DLG gives finer-grained bi-grams like $\{ri, ch, ne, ss\}$ which could be potentially important for short text modeling with small word vocabulary like text entailment task.

### Reading Comprehension

To investigate the effectiveness of the subword-augmented embedding in conjunction with more complex models, we conduct experiments on machine reading comprehension tasks. The reading comprehension task can be described as a triple $\langle D, Q, A \rangle$, where $D$ is a document (context), $Q$ is a query over the contents of $D$, in which a word or span is the right answer $A$. This task can be divided into cloze-style and query-style. The former has restrictions that each answer should be a single word and should appear in the document and the original sentence removing the answer part is taken as the query. For the query-style, the query is

![https://dumps.wikimedia.org/](https://dumps.wikimedia.org/)

---

Table 1: Accuracy on SNLI dataset. SOTA is short for state-of-the-art.

| Model                  | Dev  | Test |
|------------------------|------|------|
| Baseline (Word + Char) | 88.39| 87.61|
| Word + Char            | 88.39| 87.61|
| Word + BPE-AV          | 88.42| 88.11|
| Word + BPE-FRQ         | 88.56| 88.36|
| Word + BPE-DLG         | 88.68| 88.56|
| SOTA [Kim et al. 2018] | /    | 88.9 |

Table 2: Accuracy on CMRC-2017 dataset.

| Model                  | Dev  | Test |
|------------------------|------|------|
| Baseline (Word + Char) | 76.15| 77.73|
| Word + MM-AV           | 77.80| 77.80|
| Word + MM-DLG          | 77.30| 77.17|
| Word + BPE-FRQ         | 78.95| 78.80|
| SOTA [Cui et al. 2017b] | 77.20| 78.63|

---

[Leaderboard:](https://nlp.stanford.edu/projects/snli/)
The batch size is 64. We applied dropout between layers with a dropout rate of 0.5. The word and subword representation were 128. We used three attention layers. The GRU hidden units for the word representation are 90 and 180, respectively. We optimize the model using ADAM. The batch size is 32 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). The GRU hidden units for both the word and subword representation were 128. We applied dropout between layers with a dropout rate of 0.5. The batch size is 64.

Table 3 gives our results on CMRC-2017 dataset which shows that our Word + BPE-FRQ model outperforms all other models on the test set, even the state-of-the-art AoA Reader (Cui et al. 2017a). With the help of the proposed method, the GA Reader could yield a new state-of-the-art performance over the dataset. Different from the above text entailment task, the best subword segmentation tends to be BPE-FRQ instead of BPE-DLG. The divergence indicates that for a task like reading comprehension involving long paragraphs with a huge vocabulary high frequency words weigh more. In fact, as DLG measures every word types equally, it can be seriously biased by a lot of noise in the vocabulary. Using frequency instead of DLG can let the segmentation resist the noise by keeping concerns over those high frequency (also usually regular) words.

**Chinese Cloze-style** The dataset (Cui et al. 2017b) was used for the 1st Evaluation on Chinese Machine Reading Comprehension (CMRC-2017), which contains 350k+ queries. Our baseline model is the Gated-Attention (GA) Reader (Dhingra et al. 2017) which integrates a multi-hop architecture with a gated attention mechanism between the intermediate states of document and query.

We used stochastic gradient descent with ADAM updates for optimization. The batch size was 32 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 used three attention layers. The GRU hidden units for both the word and subword representation were 128. We applied dropout between layers with a dropout rate of 0.5. The batch size is 64.

Table 2 gives our results on CMRC-2017 dataset which shows that our Word + BPE-FRQ model outperforms all other models on the test set, even the state-of-the-art AoA Reader (Cui et al. 2017a). With the help of the proposed method, the GA Reader could yield a new state-of-the-art performance over the dataset. Different from the above text entailment task, the best subword segmentation tends to be BPE-FRQ instead of BPE-DLG. The divergence indicates that for a task like reading comprehension involving long paragraphs with a huge vocabulary high frequency words weigh more. In fact, as DLG measures every word types equally, it can be seriously biased by a lot of noise in the vocabulary. Using frequency instead of DLG can let the segmentation resist the noise by keeping concerns over those high frequency (also usually regular) words.

**English Query-style** The Stanford Question Answering Dataset (SQuAD) (Rajpurkar et al. 2016) contains 100k+ crowd sourced question-answer pairs where the answer is a span in a given Wikipedia paragraph. Our basic model is Bidirectional Attention Flow (Seo et al. 2017) and we improve it by adding a self-attention layer (Wang et al. 2017) and ELMO (Peters et al. 2018), similar to (Clark and Gardner 2018), to see whether subword could still improve more complex models.

In this model, the augmented embeddings of document and query are passed through a bi-directional GRU which share parameters, and then fed to the BiDAF model. Then, we obtain the context vectors and pass them through a linear layer with ReLU activations, followed by a self-attention layer against the context itself. Finally, the results are fed through linear layers to predict the start and end token of the answer. For the hyper-parameters, the dropout rates for the GRUs and linear layers are 0.2. The dimensions for GRU and linear layers are 90 and 180, respectively. We optimize the model using ADAM. The batch size is 32.

Table 3 shows the results on the dev set which shows that our Word + BPE-FRQ model outperforms all other models on the test set, even the state-of-the-art AoA Reader (Cui et al. 2017a). With the help of the proposed method, the GA Reader could yield a new state-of-the-art performance over the dataset. Different from the above text entailment task, the best subword segmentation tends to be BPE-FRQ instead of BPE-DLG. The divergence indicates that for a task like reading comprehension involving long paragraphs with a huge vocabulary high frequency words weigh more. In fact, as DLG measures every word types equally, it can be seriously biased by a lot of noise in the vocabulary. Using frequency instead of DLG can let the segmentation resist the noise by keeping concerns over those high frequency (also usually regular) words.

**Analyis** The experimental results have shown that the subword-augmented embedding can essentially improve baselines, from the simple to the complicated, among multiple tasks with different languages. Though the performance of BPE-FRQ tends to be the most stable overall, the best practice for subword embedding might be task-specific. This also discloses that there exists potential for a more effective goodness measure or segmentation algorithm to polish up the subword representations.

**Using Diverse Embedding Together** To see if we can receive further performance improvement when using different embedding together, we compare the following embeddings: Word Only, Char Only, BPE-FRQ only and Word + Char, Word + BPE-FRQ and Word + Char + BPE-FRQ. Table 4 shows the result. For each type of

---

Table 3: Exact Match (EM) and F1 scores on SQuAD dev set. BiDAF\(_\alpha\) denotes BiDAF + Self-Attention and BiDAF\(_\beta\) denotes BiDAF + Self-Attention + ELMO.

| Model | EM   | F1   |
|-------|------|------|
| Word + Char | 71.22 | 80.42 |
| Word + MM-AV | 72.46 | 81.28 |
| Word + MM-DLG | 72.21 | 81.03 |
| Word + BPE-FRQ | 72.79 | 81.78 |
| Word + Char | 77.43 | 83.03 |
| Word + MM-AV | 77.49 | 85.23 |
| Word + MM-DLG | 77.46 | 85.22 |
| Word + BPE-FRQ | 77.84 | 85.48 |
| Word + Char | 68.23 | 77.95 |
| Word + MM-AV | 68.86 | 78.44 |
| Word + MM-DLG | 68.82 | 78.40 |
| Word + BPE-FRQ | 69.35 | 78.97 |

Table 4: Embedding combinations on CMRC-2017.

| Model | Dev | Test |
|-------|-----|------|
| Word Only | 74.90 | 75.80 |
| Char Only | 71.25 | 72.53 |
| BPE-FRQ Only | 74.75 | 75.77 |
| Word + Char | 76.15 | 77.73 |
| Word + BPE-FRQ | 78.95 | 78.80 |
| Word + Char + BPE-FRQ | 79.05 | 78.83 |

3CMRC-2017 Leaderboard: [http://www.hfl-tek.com/cmrc2017/leaderboard/](http://www.hfl-tek.com/cmrc2017/leaderboard/)

5The word vocabulary sizes of SNLI and CMRC-2017 are 30k and 90k respectively.

6Since the test set is not released, we train our models on training set and evaluate our them on dev set.

---
embedding alone, word embedding and BPE-FRQ subword embedding turn out to be comparable. BPE-FRQ performs much better than char embedding, which again confirms that subwords are more representative as minimal natural linguistic units than single characters. Any embedding combination could improve the performance as the distributed representations can be beneficial from different perspectives through diverse granularity. However, using all the three types of embeddings only shows marginal improvement. This might indicate increasing embedding features or dimension might not bring much gains and seeking natural and meaningful linguistic units for representation is rather significant.

**Subword Vocabulary Size**

The segmentation granularity is highly related to the subword vocabulary size. For BPE style segmentation, the resulting 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 of BPE-FRQ from 0 to 20k, and conduct quantitative study on SNLI, CMRC-2017 and SQuAD for BPE-FRQ segmentation. Figure 1 shows the results. We observe that with 1k merge times, the models could obtain the best performance on CMRC-2017 and SQuAD though these two tasks are of different languages while 10k shows to be more suitable for SNLI. The results also 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.

**Subword and Word Embedding Integration Strategies**

We investigate the combination of subword-augmented embedding with word embedding. Table 5 shows the comparisons based on our best models of SNLI and CMRC-2017, BPE-DLG and BPE-FRQ, respectively. The models with concat and mul significantly outperform the model with sum. This reveals that concat and mul operations might be more informative than sum and the best practice for the choice would be task-specific. Though concat operation may result in high dimensions, it could keep more information for downstream models to select from. The superiority of mul might be due to element-wise product being capable of modeling the interactions and eliminating distribution differences between word and subword embedding which is intuitively similar to endowing subword-aware attention over the word embedding. In contrast, sum is too simple to prevent from detailed information losing.

**Table 5:** Different merging functions with word embeddings on SNLI.

| Model | Strategy | Dev  | Test |
|-------|----------|------|------|
| SNLI  | concat   | 88.68| 88.56|
|       | sum      | 88.30| 87.14|
|       | mul      | 88.47| 87.77|
| CMRC  | concat   | 77.45| 77.47|
|       | sum      | 75.95| 76.43|
|       | mul      | 78.95| 78.80|

**Figure 1:** Case study of the subword vocabulary size of BPE-FRQ.

**Figure 2:** Results of n-gram of MM-AV measurement on SQuAD dataset.
Figure 3: Pair-wise attention visualization.

Effect of the $n$-grams
The goodness measures commonly build the subword vocabulary based on neighbored character relationship inside words. This is reasonable for Chinese where words are commonly formed by two characters which is also the original motivation for Chinese word segmentation. However, we wonder whether it would be better to use longer $n$-gram connections. We expand the $n$-grams of MM-AV from 1 to 4. Figure 2 shows the quantitative study results. We observe the $n$-grams of MM-AV segmentation might slightly influence the result where 2 or 3 tend to be better choice.

Visualization
To analyze the learning process of our models, we draw the attention distributions at intermediate layers based on an example from CMRC-2017 dataset. Figure 3 shows the result of model with BPE-FRQ. We observe that the right answer (The cat) could obtain a high weight after the pair-wise matching of document and query. After attention learning, the key evidence of the answer would be collected and irrelevant parts would be ignored. This shows our subword-augmented embedding is effective at selecting the vital points at the fundamental embedding layer, guiding the attention layers to collect more relevant pieces.

Subword Observation
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 a task like cloze-style reading comprehension where the answer is only one word or phrase. To get an intuitive observation for the task, we collect all the 118 questions whose answers are OOV words (with their corresponding documents, denoted as OOV questions) from CMRC-2017 test set, and use our model to answer these questions. We observe only 2.54% could be correctly answered by the best Word + Char embedding based model. With BPE-FRQ subword embedding, 12.71% of these OOV questions could be correctly solved. This shows the subword representations could be essentially useful for modeling rare and unseen words. In fact, the meaning of complex words like indispensability could be accurately refined by segmented subwords as shown in Table 6. This also shows subwords could help the models to use morphological clues to form robust word representations.

| Word               | Subword               |
|--------------------|-----------------------|
| indispensability   | in disp ens ability   |
| intercontinentalexchange | inter contin ent al ex change |
| playgrounds        | play ground s         |
| 大花猫              | 大 花 猫               |
| 一步一个脚印        | 一 步 一 个 脚 印       |

Table 6: Examples of BPE-FRQ subwords.

Conclusion
Embedding is the fundamental part of deep neural networks, which could also be the bottleneck of the model strength. Building a more fine-grained representation at the very beginning could potentially guide the following networks, especially attention component to collect more important pieces. This paper presents a general yet effective architecture, subword-augmented embedding to enhance the word representation and effectively handle rare or unseen words. The proposed method takes variable-length subwords segmented by unsupervised segmentation measures, without re-
lying on any predefined linguistic resource. Thus the proposed method is also suitable for various open vocabulary NLP tasks. Our work discloses that the deep internals of sub-word level embeddings are crucial, helping downstream models to absorb different signals. Experiments on three datasets from text entailment and reading comprehension tasks demonstrate significant performance gains over the baselines.

References

[Avraham and Goldberg 2017] Avraham, O., and Goldberg, Y. 2017. The interplay of semantics and morphology in word embeddings. *EACL 2017*.

[Bergmanis and Goldwater 2017] Bergmanis, T., and Goldwater, S. 2017. From segmentation to analyses: a probabilistic model for unsupervised morphology induction. *EACL 2017*.

[Bojanowski et al. 2017] Bojanowski, P.; Grave, E.; Joulin, A.; and Mikolov, T. 2017. Enriching word vectors with subword information. *TACL*.

[Botha and Blunsom 2014] Botha, J. A., and Blunsom, P. 2014. Compositional morphology for word representations and language modelling. *ICML 2014*.

[Bowman et al. 2015] Bowman, S. R.; Angeli, G.; Potts, C.; and Manning, C. D. 2015. A large annotated corpus for learning natural language inference. *EMNLP 2015*.

[Cai et al. 2017] Cai, D.; Zhao, H.; Zhang, Z.; Xin, Y.; Wu, Y.; and Huang, F. 2017. Fast and accurate neural word segmentation for Chinese. *ACL 2017*.

[Cao and Rei 2016] Cao, K., and Rei, M. 2016. A joint model for word embedding and word morphology. *The Workshop on Representation Learning for NLP*.

[Chen et al. 2017] Chen, Q.; Zhu, X.; Ling, Z.; Wei, S.; Jiang, H.; and Inkpen, D. 2017. Enhanced lstm for natural language inference. *ACL 2017*.

[Cheng, Dong, and Lapata 2016] Cheng, J.; Dong, L.; and Lapata, M. 2016. Long short-term memory-networks for machine reading. *EMNLP 2016*.

[Clark and Gardner 2018] Clark, C., and Gardner, M. 2018. Simple and effective multi-paragraph reading comprehension. *ACL 2018*.

[Cui et al. 2017a] Cui, Y.; Chen, Z.; Wei, S.; Wang, S.; Liu, T.; and Hu, G. 2017a. Attention-over-attention neural networks for reading comprehension. *ACL 2017*.

[Cui et al. 2017b] Cui, Y.; Liu, T.; Chen, Z.; Ma, W.; Wang, S.; and Hu, G. 2017b. Dataset for the first evaluation on Chinese machine reading comprehension. *arXiv preprint arXiv:1511.02301*.

[Dhingra et al. 2017] Dhingra, B.; Liu, H.; Yang, Z.; Cohen, W. W.; and Salakhutdinov, R. 2017. Gated-attention readers for text comprehension. *ACL 2017*.

[Feng et al. 2004] Feng, H.; Chen, K.; Deng, X.; and Zheng, W. 2004. Accessor variety criteria for chinese word extraction. *Computational Linguistics*.

[Gage and Philip 1994] Gage, and Philip. 1994. A new algorithm for data compression. *C Users Journal*.

[Hammarstrom and Borin 2011] Hammarstrom, H., and Borin, L. 2011. Unsupervised learning of morphology. *Computational Linguistics*.

[Joulin et al. 2017] Joulin, A.; Grave, E.; Bojanowski, P.; and Mikolov, T. 2017. Bag of tricks for efficient text classification. *EACL 2017*.

[Kim et al. 2016] Kim, Y.; Jernite, Y.; Sontag, D.; and Rush, A. M. 2016. Character-aware neural language models. *AAAI 2016*.

[Kim et al. 2018] Kim, S.; Hong, J. H.; Kang, I.; and Kwak, N. 2018. Semantic sentence matching with densely-connected recurrent and co-attentive information. *Kingma and Ba 2014* Kingma, D., and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

[Li et al. 2018] Li, H.; Zhang, Z.; Ju, Y.; and Zhao, H. 2018. Neural character-level dependency parsing for Chinese. *AAAI-18*.

[Ling et al. 2015] Ling, W.; Luis, T.; Marujo, L.; Astudillo, R. F.; Amir, S.; Dyer, C.; Black, A. W.; and Trancoso, I. 2015. Finding function in form: Compositional character models for open vocabulary word representation. *EMNLP 2015*.

[Luong and Manning 2016] Luong, M.-T., and Manning, C. D. 2016. Achieving open vocabulary neural machine translation with hybrid word-character models. *arXiv preprint arXiv:1604.00788*.

[Mikolov et al. 2013] Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

[Miyamoto and Cho 2016] Miyamoto, Y., and Cho, K. 2016. Gated word-character recurrent language model. *EMNLP 2016*.

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

[Rajpurkar et al. 2016] Rajpurkar, P.; Zhang, J.; Lopyrev, K.; and Liang, P. 2016. SQuAD: 100,000+ questions for machine comprehension of text. *EMNLP 2016*.

[Sennrich, Haddow, and Birch 2016] Sennrich, R.; Haddow, B.; and Birch, A. 2016. Neural machine translation of rare words with subword units. *ACL 2016*.

[Seo et al. 2017] Seo, M.; Kembhavi, A.; Farhadi, A.; and Hajishirzi, H. 2017. Bidirectional attention flow for machine comprehension. *ICLR 2017*.

[Shannon 1948] Shannon, C. E. 1948. A mathematical theory of communication. *Bell System Technical Journal*.

[Srivastava, Greff, and Schmidhuber 2015] Srivastava, R. K.; Greff, K.; and Schmidhuber, J. 2015. Training very deep networks. *arXiv preprint arXiv:1507.06228*.

[Tan et al. 2018] Tan, C.; Wei, F.; Yang, N.; Du, B.; Lv, W.; and Zhou, M. 2018. S-net: From answer extraction to answer generation for machine reading comprehension. *AAAI 2018*.
[Verwimp et al. 2017] Verwimp, L.; Pelemans, J.; Hamme, H. V.; and Wambacq, P. 2017. Character-word LSTM language models. EACL.

[Wang et al. 2017] Wang, W.; Yang, N.; Wei, F.; Chang, B.; and Zhou, M. 2017. Gated self-matching networks for reading comprehension and question answering. ACL 2017.

[Wang et al. 2018] Wang, R.; Zhao, H.; Ploux, S.; Lu, B.-L.; Utiyama, M.; and Sumita, E. 2018. Graph-based bilingual word embedding for statistical machine translation. TALLIP 17(4):31:1–31:23.

[Wang, Hamza, and Florian 2017] Wang, Z.; Hamza, W.; and Florian, R. 2017. Bilateral multi-perspective matching for natural language sentences. IJCAI 2017.

[Wilks 1999] Wilks, Y. 1999. Unsupervised learning of word boundary with description length gain. CoNLL-99.

[Yang et al. 2017] Yang, Z.; Dhingra, B.; Yuan, Y.; Hu, J.; Cohen, W. W.; and Salakhutdinov, R. 2017. Words or characters? fine-grained gating for reading comprehension. ICLR 2017.

[Yang, Salakhutdinov, and Cohen 2016] Yang, Z.; Salakhutdinov, R.; and Cohen, W. 2016. Multi-task cross-lingual sequence tagging from scratch. arXiv preprint arXiv:1603.06270.

[Zhang and Zhao 2018] Zhang, Z., and Zhao, H. 2018. One-shot learning for question-answering in gaokao history challenge. In COLING 2018.

[Zhang et al. 2018a] Zhang, Z.; Li, J.; Zhao, H.; and Tang, B. 2018a. Sjtu-nlp at semeval-2018 task 9: Neural hypernym discovery with term embeddings. In SemEval 2018, Workshop of NAACL-HLT 2018.

[Zhang et al. 2018b] Zhang, Z.; Li, J.; Zhu, P.; and Zhao, H. 2018b. Modeling multi-turn conversation with deep utterance aggregation. In COLING 2018.

[Zhang, Huang, and Zhao 2018] Zhang, Z.; Huang, Y.; and Zhao, H. 2018. Subword-augmented embedding for cloze reading comprehension. COLING 2018 1802–1814.

[Zhao and Kit 2008] Zhao, H., and Kit, C. 2008. An empirical comparison of goodness measures for unsupervised Chinese word segmentation with a unified framework. IJCNLP 2018.