Unsupervised Learning Helps Supervised Neural Word Segmentation

Xiaobin Wang,[1] **Deng Cai,**[2*] **Linlin Li,**[1†] **Guangwei Xu,**[1] **Hai Zhao,**[3†] **Luo Si**[1]
[1]Alibaba Group, [2]The Chinese University of Hong Kong, [3]Shanghai Jiao Tong University
{xuanjie.wxb, linyan.lll, kunka.xgw, luo.si}@alibaba-inc.com
thisisjcykcd@gmail.com, zhaohai@cs.sjtu.edu.cn

Abstract

By exploiting unlabeled data for further performance improvement for Chinese word segmentation, this work makes the first attempt at exploring adding unsupervised segmentation information into neural supervised segmenter. We survey various effective strategies, including extending the character embedding, augmenting the word score and applying multi-task learning, for leveraging unsupervised information derived from abundant unlabeled data. Experiments on standard data sets show that the explored strategies indeed improve the recall rate of out-of-vocabulary words and thus boost the segmentation accuracy. Moreover, the model enhanced by the proposed methods outperforms state-of-the-art models in closed test and shows promising improvement trend when adopting three different strategies with the help of a large unlabeled data set. Our thorough empirical study eventually verifies the proposed approach outperforms the widely-used pre-training approach in terms of effectively making use of freely abundant unlabeled data.

Introduction

Chinese word segmentation (CWS) is a fundamental task for Chinese language processing because there is no explicit word boundary marker in written Chinese while other high-level tasks rely heavily on words. In the last decades, most of the work addressing the task uses supervised models, which learn segmentation model based on some training data that have been segmented manually.

Conventionally, CWS is usually treated as a sequence labeling problem where a wide range of statistical methods are applied (Xue and others 2003; Low, Ng, and Guo 2005; Peng, Feng, and McCallum 2004; Zhao, Huang, and Li 2006; Zhao et al. 2006), including Conditional Random Fields (CRFs).

Along with deep neural models widely-used for natural language processing (NLP) tasks, previous explorers working on CWS used the traditional sequence labeling framework, but instead of extracting discrete features, they used various neural networks (e.g., long short-term memory network, LSTM (Hochreiter and Schmidhuber 1997)) for automatic feature learning and discrimination (Zheng, Chen, and Xu 2013; Pei, Ge, and Chang 2014; Chen et al. 2015; Xu and Sun 2016; Chen et al. 2017; Zhang, Liu, and Fu 2018; Ma, Ganchev, and Weiss 2018; Zhao et al. 2018). Transition based (Zhang, Zhang, and Fu 2016) and semi-CRF structure based (Liu et al. 2016) neural models have also be explored. Both of them can use complete word features. (Cai and Zhao 2016) proposed a direct structured learning method modeling the segmentation-determined compositionality of sentences. It is the state-of-the-art CWS model (closed test)[1] and we follow this work as a baseline.

So far, supervised methods have achieved satisfactory performance. Yet this ability relies on large labeled corpus. Theoretically, there are always new words which can not be covered by even very huge corpora. Generally, the optimal parameters of neural models are to be obtained via training on labeled data. Therefore, these methods have strong power of disambiguation via likelihood estimation but little capacity to deal with unknown words, i.e., out-of-vocabulary (OOV). In addition, supervised methods only make use of local information about individual characters and/or n-grams(words) within a sentence, resorting to little global information derived from a large scope over the entire corpus.

In contrast, unsupervised segmentation methods do not require any labeled data for training. These approaches intend to derive segmentation model directly from unlabeled text. There are two typical manners. One is estimating the likelihood of a character sequence to be a word, and the other is estimating the probability of a segmentation given the certain sentence from the view of language model. Beyond the scope of training data, unsupervised methods can detect OOVs continuously along with the expansion of data. This may be an important auxiliary to supervised methods. Hence, we explore the role of unsupervised segmentation in term of helping neural CWS.

The CWS benchmark evaluation, SIGHAN-Bakeoff shared task (Emerson 2005), distinguished the official CWS evaluation into two types, closed test and open test. The former only allows standard training set for segmenter learning,

*The work was conducted when Deng Cai was an intern of Alibaba from Shanghai Jiao Tong University.

†Corresponding authors.

[1]bcmi.sjtu.edu.cn/service/cws_rankings/

	Unsupervised CWS	Pretraining
non-NN	Zhao & Kit (2008) Zhao & Kit (2011)	-
NN	**this work**	Yang (2017) Zhou (2017)

Table 1: Research role position of our work. NN denotes neural network based model.

while the latter has no such a restriction and any external linguistic resources in addition to the training set can be used as well. In general, the closed test setting is suitable for evaluating the strength of the model its own, and the open test setting focuses on exploring better ways of exploiting extra resources. Following the strict definition of SIGHAN-Bakeoff, using unlabeled data should belong to the open test type. However, we argue that even within the type, using unlabeled resource such as abundant plain text and labeled (annotated) resources such as segmentation corpus or carefully selected lexicon should be also distinguished, as labeled and unlabeled data may have quite different availability and they come at different expenses. Using unlabeled data surely shows more flexibility and more adaptation when considering that unlabeled data may be freely obtained without any limitation. For neural CWS models, exploiting unlabeled data can be naturally implemented through pretrained character or word embedding (Yang, Zhang, and Dong 2017; Zhou et al. 2017). However, the pretrained embedding may still has its limitation, and many more other ways of exploiting unlabeled data are worth exploring. This work thoroughly studies ways of utilizing unsupervised segmentation over the unlabeled data to improve neural CWS models.

To our best knowledge, there has not yet been any comprehensive evaluation on strengthening neural segmentation models with unlabeled data by means of unsupervised segmentation methods. This study proposes to incorporate unsupervised segmentation information into neural models at different levels, whose research role position is illustrated in Table 1. Our experimental results show that unsupervised segmentation indeed boosts the recall rate of out-of-vocabulary by up to 4.7%. The overall segmentation performance gains an improvement of 0.6% in F_1-score. We also show that the performance improves along with the growth of unlabeled data size.

The rest of this paper is organized as follows: Firstly, we briefly review some related works; Secondly, we introduce the framework of injecting unsupervised segmentation into neural word segmentation models; Thirdly, we evaluate the proposed methods on both closed and open setting; Finally, we draw some conclusions.

Related Work

Our proposed model involves both unsupervised segmentation models and neural segmentation models. This section gives a brief introduction to these models.

Recently, neural models sprang up and have been widely studied in CWS task. Many of them take CWS problem as a tagging task, which scores tags on individual characters. Several neural architectures (Zheng, Chen, and Xu 2013; Pei, Ge, and Chang 2014; Chen et al. 2015; Ma and Hinrichs 2015; Xu and Sun 2016) have been explored. (Zheng, Chen, and Xu 2013) firstly adopted Convolution Neural Network based model to CWS problem and obtained comparable performance against basic CRF models. (Pei, Ge, and Chang 2014) proposed the Max-Margin Tensor Neural Network and introduced the tag embedding as input to capture the combinations of context and tag history. (Chen et al. 2015) utilized LSTM to keep the longer history information in memory cell and avoid the limitation of window-based approaches.

Besides, some alternatives to sequence labeling model exist. Beyond character information, they leverage word level information by means of transition-based model (Zhang, Zhang, and Fu 2016) and semi-CRF based model (Liu et al. 2016). (Cai and Zhao 2016; Cai et al. 2017) proposed to score candidate segmented outputs directly by modeling the segmentation process.

Unsupervised word segmentation methods learn segmentation models from unsegmented text data. Heretofore, these methods in literature can be roughly classified into two categories, goodness measure based methods and statistical language model based methods (Chen, Chang, and Pei 2014). The former assigns each character n-gram a score, namely, goodness measure, which indicates the likelihood to be a word. Goodness measures including description length gain (DLG) (Kit and Wilks 1999) accessor variety (AV) (Feng et al. 2004), boundary entropy (BE) (Jin and Tanaka-Ishii 2006) are popular in previous works. Then, decoding approaches like Viterbi algorithm are applied to find the best segmentation by maximizing the sum of word scores. The latter (language model based method) scores one segmentation from the view of language model. Given a sequence of characters, these methods obtain the best segmentation decision by choosing the candidate of highest posterior probability which is based on statistic language models, typically, Hierarchical Dirichlet Process (HDP) (Goldwater, Griffiths, and Johnson 2009) and Nested Pit-Yor Language Model (NPYLM) (Mochihashi, Yamada, and Ueda 2009).

To improve conventional CWS model with unsupervised segmentation, (Zhao and Kit 2011) carefully designed some features to inject goodness scores into CRF model. (Sun and Xu 2011) went a step further, other than using statistic, they also introduced natural boundary information like punctuation. (Fujii, Domoto, and Mochihashi 2017) proposed a semi-supervised model which combines NPYLM and CRF. Nevertheless, how to utilize unsupervised approaches to enhancing neural word segmentation has not been studied yet, as far as we know.

Pretraining methods are related works from the view of accessing external resources in an open test setting. It is an effective way to inherit information from extra data. (Yang, Zhang, and Dong 2017) utilized various kinds of data including unlabeled data, auto-segmented data and POS data. (Zhou et al. 2017) proposed a novel word context character embedding to benefit from automatically labeled data. However, our method only uses unlabeled data, which can

be obtained more easily than resources like POS data. Besides, extra labeled data is not necessarily needed to train the segmenter for generating auto-segmented data. In other words, our method is also suitable for closed test setting.

Methodology

This work proposes a framework to inject unsupervised segmentation information into neural models for CWS. The framework consists of two parts, unsupervised segmentation results (intermediate or final) as auxiliary information and neural segmentation model as a backbone model. Before presenting the methods, we introduce the unsupervised segmentation methods we used and the basic neural word segmentation model first.

Unsupervised Word Segmentation

We make use of both kinds of unsupervised methods mentioned in related work, namely, goodness measure based methods and statistical language model based methods.

Goodness measure based methods assign each character n-gram a goodness measure indicating the likelihood to be a word. A viterbi algorithm is applied to find the best segmentation by maximizing the sum of goodness measure. The proposed method only includes the goodness measure rather than the final segmentation (See the next two sections). Among the three goodness measure, we choose accessor variety (AV) since previous work concluded its effectiveness in improving word segmentation. This approach also gained the best results on our development set. Accessor variety counts the distinct neighbours of a character sequence. Intuitively, characters in a word co-occur frequently but their neighbours vary. A character sequence with larger AV is more likely to be a word. Similar to (Zhao and Kit 2011), we apply $log(\cdot)$ to smooth the difference in AV and do discretization to act as normalization.

$$discrete\_AV(s) = \lfloor log(AV(s)) \rfloor$$

Goodness measure is independent to specific context, so it can hardly deal with ambiguity which is easier for language model based models as they consider the whole sentence. In this work, we make use of the Nested Pit-Yor Language Model (NPYLM) (Mochihashi, Yamada, and Ueda 2009), which is a hierarchical Bayesian language model trained with Markov Chain Monte Carlo and decodes with sentence-wise Gibbs sampling. The word "hierarchical" means that the model consists of two language models: One is a character-level model for estimating the probability that a character sequence becomes a word; The other is a word-level language model for estimating the probability that a word appears after a sequence of words. It showed the state-of-the-art performance of single model unsupervised segmentation.

Neural Word Segmentation

As for Neural Word Segmentation model, we take (Cai et al. 2017) as a baseline model, since it so far gives the best closed test results on the standard benchmark dataset. This adopted model takes the segmentation process as two stages of encoder from character, word to sentence. Figure 1 illustrates the model (except for the gray parts.).

Word Encoder For a possible segmentation $w_1, w_2, \ldots, w_m$ given an input character sequence $c_1, c_2, \ldots, c_n$, where $w_i = c_{i_b}, \ldots, c_{i_e}$ (i_b and i_e index the starting point and end point of word w_i respectively), the word encoder first coverts the word sequence into a sequence of character-aware embedding, and then uses a neural network $\boldsymbol{w}_i = f(\boldsymbol{c}_{i_b}, \cdots, \boldsymbol{c}_{i_e})$ [2] to compute the vector representation of corresponding words. Based on the vector, word score is calculated to model the likelihood that w_i is a soundness word. The calculation is formed as (1), where $\boldsymbol{u}$ is a trainable parameter.

$$s^w(\boldsymbol{w}_i) = \boldsymbol{w}_i \cdot \boldsymbol{u} \tag{1}$$

Sentence Scorer To score a candidate segmentation at sentential level, the model uses linking score to model the fluency of word sequence. The generated word vector sequence is then fed into a recurrent neural network (RNN) as in its order. At each time step i, a prediction $\boldsymbol{p}_i$ for next word is produced base on the current hidden state of the RNN $\boldsymbol{h}_i$. Linking score is the dot product between the word representation and corresponding prediction. The score of a sentence is the sum of all word scores and linking scores. Formally, the score function $s(\cdot)$ is defined as the following:

$$\boldsymbol{h}_i = \text{LSTM}(\boldsymbol{h}_{i-1}, \boldsymbol{w}_i) \tag{2}$$

$$\boldsymbol{p}_i = \tanh(\boldsymbol{W}_h \boldsymbol{h}_i + \boldsymbol{b}_h) \tag{3}$$

$$s_i^l = {\boldsymbol{p}_{i-1}}^{\mathrm{T}} \boldsymbol{w}_i \tag{4}$$

$$s(\boldsymbol{w}_1, \cdots, \boldsymbol{w}_m) = \textstyle\sum_{i=1}^{m} (s_i^l + s^w(\boldsymbol{w}_i)) \tag{5}$$

The decoding algorithm [2] is implemented as beam searching for the highest-scored segmentation as output.

Utilizing Unsupervised Segmentation in Supervised Segmentation

We investigate three strategies to make the baseline model benefit from unsupervised segmentation results. First, encode an unsupervised segmentation result by label embedding to extend the character embedding. Second, enhance the word score with the goodness measure. Third, expand the training data with unsupervised segmented data and train the model in the manner of multi-task learning.

Add Label Embedding As mentioned in Section Word Encoder, characters of input sentence are usually represented as distributional embedding and thus can be fed into neural models. Given no extra input is available, the character embedding is the major information which the model depends on to make segmentation decision. Hence, it is considerable to enrich the character embedding to make it carry more prior knowledge into the model.

[2] We refer the interesting readers to see more details in (Cai et al. 2017). w denotes word/word embedding indiscriminately. So is the symbol c.

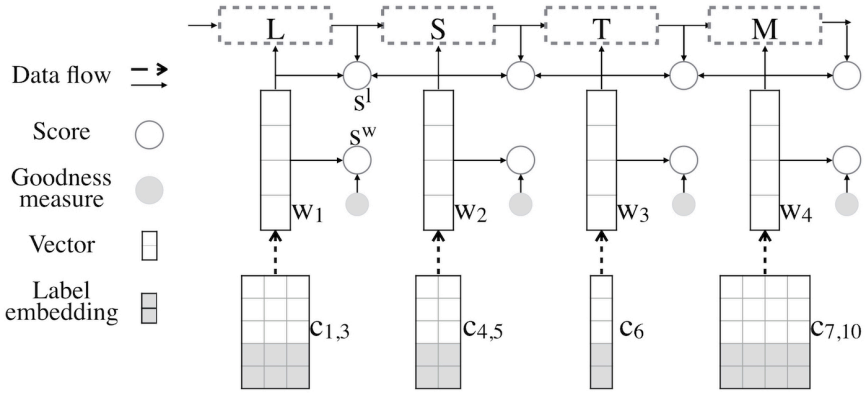

Figure 1: The baseline model augmented with the proposed methods, where $c_{i,j}$ denotes character embedding and w_i denotes word vector. s^w and s^l denote word score and linking score respectively. The gray parts are the unsupervised information. Dash lines mean shared parts in multi-task learning.

Our framework extends the character embedding with label embedding to blend in unsupervised information, as demonstrated using gray lattices in Figure 1. Input sentences are segmented in advance by an unsupervised segmentation model which is learned from the same dataset but with segmentation boundaries removed. Then, the segmented results are interpreted into label sequences. Each character in the input sentence corresponds to one of the labels in {B, I, E, S}, where B, I, E indicate the beginning, the middle, and the end of a word respectively, and S indicates a single character word. We introduce another embedding layer for these labels. Therefore, the model takes the concatenation of character embedding and label embedding as input. The new word encoder is formalized as:

$$\boldsymbol{w}_i = f(\boldsymbol{c}_{i_b} \oplus \boldsymbol{l}_{i_b}, \cdots, \boldsymbol{c}_{i_e} \oplus \boldsymbol{l}_{i_e})$$

where symbol $\oplus$ denotes vector concatenation and $\boldsymbol{l}$ denotes embedding of labels. Intuitively, the embedding of labels serves as the prior knowledge of word boundaries for the neural model.

Augment Word Score The baseline model introduces the word score (white circles in Figure 1) to estimate the likelihood that a sequence of characters is a soundness word. Similarly, the aforementioned goodness measure (gray circles in Figure 1) plays the same role to evaluate a word from a corpus-level point of view. It makes sense to enhance the word score by goodness measure in order to consider both local and global information. Instead of linear combination with fixed weight, we use the following formulas to merge these two values, where, $\boldsymbol{w}$ is the word vector; $\boldsymbol{\alpha}$ is a weight value calculated as the dot product of $\boldsymbol{w}$ and a trainable vector $\boldsymbol{v}$. Such an adaptive weight $\boldsymbol{\alpha}$ enlarges the model capacity and avoids under-fitting.

$$s^{w'}(\boldsymbol{w}) = s^w(\boldsymbol{w}) + \alpha \cdot goodness(w)$$
$$\alpha = \boldsymbol{v} \cdot \boldsymbol{w}$$

Apply Multi-task Learning For an overall-level knowledge distillation from unsupervised segmentation, we also use a multi-task learning strategy (Luong et al. 2015), i.e., to train our neural model by both human-annotated gold data and the natural segmented results by the unsupervised approach. The two tasks are learned simultaneously with shared bottom layers, where the knowledge transfer occurs. Specifically, as illustrated in Figure 1 by dash lines, the word encoder and the LSTM for scoring words are shared across tasks, while others are trained to fit on different datasets. Formally, formula (2), (5) are not changed, while formula (1), (3), (4) are modified to:

$$s^w(\boldsymbol{w}_i) = \boldsymbol{w}_i \cdot \boldsymbol{u}^m$$
$$\boldsymbol{p}_i^m = \tanh(\boldsymbol{W}_{(h)}^m \boldsymbol{h}_i + \boldsymbol{b}_h^m)$$
$$s_i^{l,m} = {\boldsymbol{p}_{i-1}^m}^{\mathrm{T}} \boldsymbol{w}_i, m \in \{0, 1\}$$

where variables marked with m are isolated between two tasks and m indicates the task id. [3] Algorithm 1 gives the detailed learning process.

Strategy Collaboration Aiming at taking advantage of different unsupervised methods and explored strategies, we use them in combination. Each unsupervised method has its own strength. Goodness measure is strong at generating high quality candidate word list, but is subject to ambiguity, which is a main factor for errors occurring in segmentation (Huang and Zhao 2007). In contrast, language model based approaches can generate more accurate segmentation although the maximum word length is limited. Therefore, we make use of the language model NPYLM for adding label embeddings and for multi-task learning, and goodness measure for augmenting word score. Moreover, we combine label embedding and word score (or multi-task learning and word score) to make use of different unsupervised approaches.

[3] We also tried Generative Adversarial Network, but the experiments did not show positive results.

Algorithm 1 multi-task learning with unlabeled data

Input: labeled data (x_i, y_i), $i \in (1, m)$, marked as task 1
unlabeled data (x_i) $i \in (m+1, n)$, marked as task 2
Output: word encoder f
word score parameter u^1, u^2
LSTM parameters ϕ
linking score parameters W_h^1, b_h^1, W_h^2, b_h^2
for e from 0 to epochs **do**
  for i in batch(e) **do**
    if task($x[i]$) is 1 **then**
      y'_i = Decode($x[i]$,f, u^1, W_h^1, b_h^1, ϕ)
      loss += Compute_loss(y_i, y'_i)
    else
      $\dot{y}_i$ = Unsupervised_segment($x[i]$)
      y'_i = Decode($x[i]$, f, u^2, W_h^2, b_h^2, ϕ)
      loss += Compute_loss($\dot{y}_i$, y'_i)
    end if
  end for
  Update_parameters(loss)
end for
return f, u^1, u^2, W_h^1, b_h^1, W_h^2, b_h^2, ϕ

LE		WS		MT	
GM	NPY	GM	NPY	GM	NPY
	√	√			
		√			√
	√				√
	√	√			√

Table 2: Combination of strategies (each line denotes one combination). label embedding (LE), augmenting word score (WS), multi-task learning (MT), goodness measure (GM), and using data segmented by NPYLM (NPY).

Every explored strategy has weaknesses. Label embedding and goodness measure only consider the unsupervised information at certain layer, thus the information has less effect on the entire model. Although multi-task learning makes use of unsupervised results in many components of the model by shared parameters, it cannot obtain the case specific unsupervised segmentation information at the decoding process. Thus, multi-task learning is performed with label embedding and/or goodness measure as extra input. The detailed combination settings are listed in Table 2.

Experiments

Dataset and Setting

We evaluate the effectiveness of our methods by F_1-score on the widely used benchmark datasets, i.e., PKU, MSR, AS and CITYU, from the 2nd international CWS Bakeoff (Bakeoff-2005) (Emerson 2005). The former two are written in simplified Chinese while the latter two in traditional Chinese. The statistics of the data are listed in Table 3.

The baseline model implementation is cloned from Github for the baseline segmenter [4]. Table 4 shows the hyper-parameters of the model. We used an open-source version of NPYLM based segmenter[5] as the unsupervised segmenter, which generates segmented texts for the label embedding and multi-task learning approaches.

		MSR	PKU	AS	CITYU
Train	#s	78k	17k	638k	48k
	#w	2,122k	1,010k	4,904k	1,310k
Dev	#s	8.7k	1.9k	71k	5.3k
	#w	246k	100k	545k	146k
Test	#s	4.0k	1.9k	14k	1.4k
	#w	106k	104k	123k	41k

Table 3: Statistics of the dataset, number of sentences (#s) and words (#w).

Parameter name	value
Character embedding size	100
Word embedding size	50
Hidden unit number	50
Margin loss discount	0.2
Maximum word length	6
Decoding beam size	1

Table 4: Hyper-parameters of the baseline model.

Parameter Selection

Apart from the hyper-parameters of the baseline model, there are two other parameters that are of study interest, i.e., the type of goodness measure and the maximum length of the word in unsupervised segmentation results. We compared some settings in this section. For efficiency, we set the maximum word length allowed in the decoding process of baseline model to be 4 and only conduct experiments on MSR and PKU. As words longer than four characters are rare in the corpus, limiting the length of words has little impact on our conclusion.

Comparing different word length Unsupervised segmentation methods like NPYLM usually have a limit for maximum word length to restrict the decoding searching space of segmentation results for computational efficiency. Table 5 demonstrates the effect of this limitation on the proposed strategies respectively. The maximum word length we tried is 4 characters since the baseline model restricts the word length to be less than 4. We train the NPYLM model on the combination of the training set and the test set without segmentation labels.

As shown in Table 5, limiting the number of characters in a word to be no more than 2 achieves the best result (the remaining experiments adopt this setting). By comparing these segmentation results, we find that longer limit of word length leads to many phrases in results. These phrases often consist of three or more characters and can be further segmented to fine-grained words according to the annotation schema of

[4] https://github.com/jcyk/greedyCWS

[5] https://github.com/musyoku/python-npylm

Integration	LE		MT	
Len.	MSR	PKU	MSR	PKU
2	97.1	95.5	97.1	95.5
3	97.1	95.5	96.9	95.5
4	97.0	95.4	96.9	95.3

Table 5: Comparing the effect of maximum word length

	MSR		PKU	
Type	orig.	disc.	orig.	disc.
AV	97.0	97.0	95.2	95.5
BE	96.9	97.0	95.0	95.3
DLG	97.0	97.0	95.0	95.3

Table 6: Comparing the effect of different goodness measures, original (orig.) and discretized (disc.).

MSR/PKU. Thus, allowing longer words has negative effect on the final neural model.

Comparing goodness measure We tested three popular goodness measures, i.e., description length gain (DLG), accessor variety (AV) and boundary entropy (BE). We experimented with both the original and the discretized goodness measure value approaches. As shown in Table 6, discretization has a positive effect, since applying $log(\cdot)$ smooths the difference in values and discretization acts as a normalization. Moreover, AV value is the most suitable method for augmenting the baseline model. As shown on the table, the performance on MSR is more stable. The main reason is that goodness measure requires sufficient appearance of words to obtain significant statistics. The MSR data has a special annotation schema which includes too many long-tailed uncommon words (Zhao et al. 2010). Hence, given abundant unlabeled data, the performance on MSR will be improved. This is verified by the experiments in the following section.

Main Results

Comparing to baseline The proposed methods are compared to the baseline in Table 7 by closed setting. In these experiments, goodness (AV) and NPYLM models are obtained based on the combination of training and test data without segmentation labels. According to Table 5, we use the optimal maximum word length of NPYLM, i.e., 2 characters.

It can be seen from Table 7 that the proposed methods can make use of the unsupervised segmentation to improve the performance of the state-of-the-art baseline modelof closed testing. According to the standard significant test criterion for CWS (Emerson 2005), the improvements are significant (at 95% confidence level). The performance gain may be attributed to the OOV problem alleviated by effectively capturing global information of unsupervised segmentation methods. In fact, we found that using strategy collaboration can provide a 4.7% boost of OOV recall rate on MSR.

As described in Section Strategy Collaboration, employing individual strategy in isolation has some drawbacks. We proposed to combine them. Comparing the last four rows and the first row (baseline) in Table 7, we can see that method combination indeed improves the performance in both F_1-score and OOV recall rate. The collaboration strategy makes the neural segmentation model learn from unsupervised segmentation results during training (via MT) as well as decoding (via LE & WS). The former helps to memorize some OOV words recognized by the unsupervised methods. The latter emphasizes words learned in unsupervised segmentation results to enable the model to make effective decisions, even the word does not appear in presegmented training corpus.

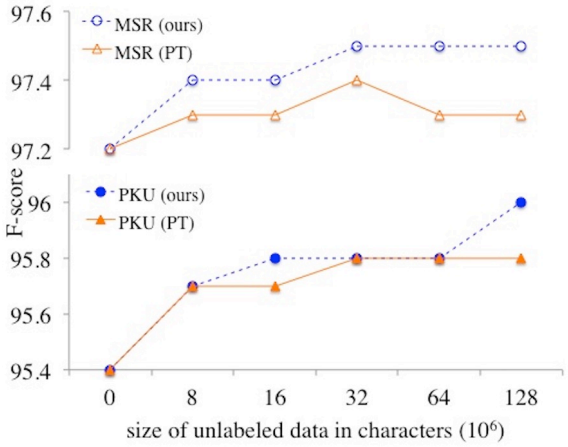

Figure 2: Performance of different sizes of unlabeled corpus. PT denotes pretraining.

Performance as the size of data grows In order to find out the potentiality of utilizing unlimited unlabeled data, we evaluate the baseline model enhanced with our proposed method (WS + LE + MT) with different sizes of external data. This evaluation is an open test setting. The Gigaword corpus (Graff and Chen 2005), an unlabeled dataset, is used as the unlabeled dataset. We did not use the entire Gigaword, nearly 352 millions of characters, because the training process of NPYLM on such a large dataset is too time consuming whereas we only aim to show the trend of performance change as the size of data increases.

Figure 2 demonstrates the relation between F_1-score and scale of the data being used. The performance of our proposed model improves steadily as the size of the data grows. This is intuitive as more unlabeled data results in better unsupervised model which is a fundamental building block of our model. We argue that, in open test, it makes more sense to evaluate the performance of the model by the trend of growth other than the absolute gain, as open test setting allow unlimited extra linguistic resources for exploration. Considering that the unlabeled texts come with low cost, our proposed strategies are of great practical value.

The performance increase of MSR is relatively modest. The reason is two fold: First, the MSR data has a special annotation schema which includes many long words. We noticed that the error of the baseline model is mainly caused

Model	MSR		PKU		AS		CITYU	
	F	R_{oov}	F	R_{oov}	F	R_{oov}	F	R_{oov}
baseline	97.2	53.7	95.4	58.5	95.4	63.1	95.6	74.2
+word score (WS)	97.3	55.8	95.4	60.2	95.4	62.6	95.7	**76.8**
+label emb. (LE)	97.3	56.4	95.6*	60.4	95.5	65.0	95.8*	74.6
+multi-task (MT)	97.3	56.1	95.5	61.0	95.4	**66.0**	95.6	76.1
WS+MT	97.2	56.7	95.6*	62.0	95.4	63.3	**95.9***	76.4
WS+LE	97.3	55.0	95.6*	60.9	95.5	63.7	**95.9***	75.0
LE+MT	**97.4***	**58.4**	**95.7***	**62.7**	95.4	63.4	95.8*	76.0
WS + LE + MT	**97.4***	**58.4**	95.6*	60.7	**95.6***	65.6	**95.9***	76.5

Table 7: Performance of different approaches of using unsupervised segmentation (%). The asterisks indicate that the improvements are significant (at 95% confidence level), following the standard significant test criterion for CWS (Emerson 2005).

by missing these long words as the model restricts the word length; Second, the NPYLM approach mainly predicts short words while excluding most long words.

Comparing to pretraining Pretraining (Yang, Zhang, and Dong 2017; Zhou et al. 2017) is a commonly employed strategy for neural segmentation model. In these methods, character embedding of the input layer is initialized by vectors trained previously on large external corpora. We also apply pretraning technique to the baseline model as the size of the data grows. The character embedding is pretrained on the Gigaword corpus by (Zhou et al. 2017), which learns character embedding on a word-based context. This model yields higher relative performance gain and is model-independent compared to other pretraining approaches.

As shown in Figure 2, the F_1-score of the model with pretraining does not constantly increase after reaching to certain level of performance as more and more data are added. Figure 2 also confirms that our proposed method outperforms the pretraining method when the size of unlabeled data scales up. Unsupervised segmentation methods are more effective in capturing word boundary explicitly, which in turn can be coupled with advanced techniques learnt on labeled data. Thus, our approach are more effective at utilizing large amount of unlabeled data than pretraining based approaches. We also combine the pretraining method with our approach to see if further improvement can be achieved. However, the gain is marginal as Table 8.

Comparing to tri-training Aiming at leveraging unlabeled data to improve the performance of supervised model with limited amount of training data, a traditional semi-supervised algorithm, tri-training, relies on three basic models to predict the initial label and takes consistent cases as extra training data (Zhou and Li 2005). To compare with tri-training, we utilized two other statistical model based segmenters, ZPar (Zhang and Clark 2007) and THULAC (Sun et al. 2016), together with the baseline model, to perform the tri-training approach. ZPar and THULAC are first trained from the same Bakeoff data as the baseline model. Then, they are applied to segment the Gigaword text. Sentences segmented consistently are used as supplementary training data of the baseline model. To reduce the training time, we keep the scale of the extra training data the same as Bakeoff training set by randomly sampling in each iteration. The results in Table 8 indicate that the tri-training method does not improve the baseline.

Model	MSR	PKU
baseline	97.2	95.4
+tri-training	97.2	95.5
+pretraining (PT)[6]	97.4	95.8
+ours best	**97.5**	96.0
+ours best + PT	**97.5**	**96.1**

Table 8: Compare with approaches using unlabeled data. As the extra data is simplified Chinese, we only do experiments on MSR and PKU.

Conclusion

This paper makes the first attempt to explore incorporating the results of unsupervised segmentation with neural word segmentation models. We extend the character embedding with the embedding of segmentation labels from unsupervised model, augment the word score with goodness measure and use multi-task learning to learn from unsupervised segmentation. Our empirical evaluation and comparisons on benchmark datasets show that our proposed methods boost the OOV recall rate significantly and outperform the baseline both in closed and open tests. In addition, we have shown that the performance is continuously improving while the size of unlabeled data increases. Furthermore, our methods have clear advantage over the pretraining method.

Acknowledgments

This work was supported by Alibaba Group through Alibaba Research Fellowship. Hai Zhao 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).

References

Cai, D., and Zhao, H. 2016. Neural word segmentation learning for Chinese. In *ACL*.

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

Chen, X.; Qiu, X.; Zhu, C.; Liu, P.; and Huang, X. 2015. Long short-term memory neural networks for Chinese word segmentation. In *EMNLP*.

Chen, X.; Shi, Z.; Qiu, X.; and Huang, X. 2017. Adversarial multi-criteria learning for Chinese word segmentation. In *ACL*.

Chen, M.; Chang, B.; and Pei, W. 2014. A joint model for unsupervised Chinese word segmentation. In *EMNLP*.

Emerson, T. 2005. The second international Chinese word segmentation bakeoff. In *SIGHAN*.

Feng, H.; Chen, K.; Kit, C.; and Deng, X. 2004. Unsupervised segmentation of Chinese corpus using accessor variety. In *IJCNLP*.

Fujii, R.; Domoto, R.; and Mochihashi, D. 2017. Nonparametric bayesian semi-supervised word segmentation. *Transactions of the Association of Computational Linguistics*.

Goldwater, S.; Griffiths, T. L.; and Johnson, M. 2009. A bayesian framework for word segmentation: Exploring the effects of context. *Cognition*.

Graff, D., and Chen, K. 2005. Chinese gigaword. *LDC Catalog No.: LDC2003T09, ISBN*.

Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural computation*.

Huang, C., and Zhao, H. 2007. Chinese word segmentation: A decade review. *Journal of Chinese Information Processing*.

Jin, Z., and Tanaka-Ishii, K. 2006. Unsupervised segmentation of Chinese text by use of branching entropy. In *ACL-COLING*.

Kit, C., and Wilks, Y. 1999. Unsupervised learning of word boundary with description length gain. *EACL-CoNLL*.

Liu, Y.; Che, W.; Guo, J.; Qin, B.; and Liu, T. 2016. Exploring segment representations for neural segmentation models. In *IJCAI*.

Low, J. K.; Ng, H. T.; and Guo, W. 2005. A maximum entropy approach to Chinese word segmentation. In *SIGHAN*.

Luong, M.-T.; Le, Q. V.; Sutskever, I.; Vinyals, O.; and Kaiser, L. 2015. Multi-task sequence to sequence learning. *arXiv preprint arXiv:1511.06114*.

Ma, J., and Hinrichs, E. 2015. Accurate linear-time Chinese word segmentation via embedding matching. In *ACL-IJCNLP*.

Ma, J.; Ganchev, K.; and Weiss, D. 2018. State-of-the-art Chinese word segmentation with bi-lstms. In *EMNLP*.

Mochihashi, D.; Yamada, T.; and Ueda, N. 2009. Bayesian unsupervised word segmentation with nested pitman-yor language modeling. In *ACL-IJCNLP*.

Pei, W.; Ge, T.; and Chang, B. 2014. Max-margin tensor neural network for Chinese word segmentation. In *ACL*.

Peng, F.; Feng, F.; and McCallum, A. 2004. Chinese segmentation and new word detection using conditional random fields. In *COLING*.

Sun, W., and Xu, J. 2011. Enhancing Chinese word segmentation using unlabeled data. In *EMNLP*.

Sun, M.; Chen, X.; Zhang, K.; Guo, Z.; and Liu, Z. 2016. Thulac: An efficient lexical analyzer for Chinese. Technical report.

Xu, J., and Sun, X. 2016. Dependency-based gated recursive neural network for Chinese word segmentation. In *ACL*.

Xue, N., et al. 2003. Chinese word segmentation as character tagging. *Journal of Computational Linguistics & Chinese Language Processing*.

Yang, J.; Zhang, Y.; and Dong, F. 2017. Neural word segmentation with rich pretraining. In *ACL*.

Zhang, Y., and Clark, S. 2007. Chinese segmentation with a word-based perceptron algorithm. In *ACL*.

Zhang, Q.; Liu, X.; and Fu, J. 2018. Neural networks incorporating dictionaries for Chinese word segmentation. In *AAAI*.

Zhang, M.; Zhang, Y.; and Fu, G. 2016. Transition-based neural word segmentation. In *ACL*.

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

Zhao, H., and Kit, C. 2011. Integrating unsupervised and supervised word segmentation: The role of goodness measures. *Information Sciences*.

Zhao, H.; Huang, C.-N.; Li, M.; and Lu, B.-L. 2006. Effective tag set selection in Chinese word segmentation via conditional random field modeling. In *PACLIC*.

Zhao, H.; Huang, C.-N.; Li, M.; and Lu, B.-L. 2010. A unified character-based tagging framework for Chinese word segmentation. *ACM Transactions on Asian Language Information Processing*.

Zhao, L.; Zhang, Q.; Wang, P.; and Liu, X. 2018. Neural networks incorporating unlabeled and partially-labeled data for cross-domain Chinese word segmentation. In *IJCAI*.

Zhao, H.; Huang, C.-N.; and Li, M. 2006. An improved Chinese word segmentation system with conditional random field. In *SIGHAN*.

Zheng, X.; Chen, H.; and Xu, T. 2013. Deep learning for Chinese word segmentation and pos tagging. In *EMNLP*.

Zhou, Z.-H., and Li, M. 2005. Tri-training: Exploiting unlabeled data using three classifiers. *IEEE Transactions on knowledge and Data Engineering*.

Zhou, H.; Yu, Z.; Zhang, Y.; Huang, S.; Dai, X.; and Chen, J. 2017. Word-context character embeddings for Chinese word segmentation. In *EMNLP*.