TopWORDS-Seg: Simultaneous Text Segmentation and Word Discovery for Open-Domain Chinese Texts via Bayesian Inference

Changzai Pan
Center for Statistical Science & Department of Industrial Engineering, Tsinghua University
pcz18@mails.tsinghua.edu.cn

Maosong Sun
Department of Computer Science and Technology & Guo Qiang Institute for Artificial Intelligence, Tsinghua University
sms@tsinghua.edu.cn

Ke Deng*
Center for Statistical Science & Department of Industrial Engineering, Tsinghua University
kdeng@tsinghua.edu.cn

Abstract

Processing open-domain Chinese texts has been a critical bottleneck in computational linguistics for decades, partially because text segmentation and word discovery often entangle with each other in this challenging scenario. No existing methods yet can achieve effective text segmentation and word discovery simultaneously in open domain. This study fills in this gap by proposing a novel method called TopWORDS-Seg based on Bayesian inference, which enjoys robust performance and transparent interpretation when no training corpus and domain vocabulary are available. Advantages of TopWORDS-Seg are demonstrated by a series of experimental studies.

1 Introduction

Due to absence of word boundaries in Chinese, Chinese natural language processing (CNLP) faces a few unique challenges, including text segmentation and word discovery. When processing open-domain Chinese corpus containing many unregistered words and named entities, these challenges become more critical as they often entangle with each other: we usually cannot segment Chinese texts correctly without knowing the underlying vocabulary; on the other hand, it is often difficult to precisely discover unregistered words and named entities from open-domain corpus without guidance on text segmentation.

Most methods for CNLP in the literature assume that the underlying vocabulary is known and focus on improving performance of text segmentation in closed test. The first category of methods along this research line are simple methods based on *Word Matching* (Chen and Liu, 1992; Geutner, 1996; Chen, 2003; Shu et al., 2017), which segment a Chinese sentence by matching sub-strings in the sentence to a pre-given vocabulary in a forward or reserve order. The second category of methods utilize manually segmented corpus or large-scale pre-training corpus to train statistical models such as *Maximum Entropy* (Berger et al., 1996; McCallum et al.; Low et al., 2005), HMM (Sproat et al., 1994; Zhang et al., 2003) and CRF (Lafferty et al., 2001; Xue, 2003; Peng et al., 2004; Luo et al., 2019), or deep learning models including CNN (Wang and Xu), LSTM (Chen et al., 2015), Bi-LSTM (Ma et al., 2018) and BERT (Yang, 2019), or hybrid models like Bi-LSTM-CRF (Huang et al., 2015) and LSTM-CNNs-CRF (Ma and Hovy, 2016), to achieve text segmentation directly or indirectly. Methods of this category have led to popular toolkits for processing Chinese texts, including Jieba (Sun, 2012), StanfordNLP (Manning et al., 2014), THULAC (Sun et al., 2016), PKUSEG (Luo et al., 2019), and LTP (Che et al., 2021). A popular strategy adopted by some of these toolkits is to segment the target texts into sequences of basic words first, and capture unregistered words and named entities, which are often word compounds consisting of basic words, later via chunking and syntactic analysis. Although such a strategy can equip these toolkits with some ability on word discovery, it is apparently sub-optimal, because we may mis-segment basic words at the first place without realizing the existence of potential technical words, making it impossible to discover technical word compounds correctly in post analysis such as chunking and syntactic analysis.

On the other hand, unsupervised methods are also developed to achieve text segmentation when no pre-given vocabulary and manually segmented training corpus are available. Some methods of this research line segment texts based on local statistics of the target texts, including *Description Length Gain* (Kit and Wilks, 1999), *Mutual Information* (Chang and Lin, 2003), *Accessor Variety* (Feng et al., 2004), *Evaluation-Selection-Adjustment Process* (Wang et al., 2011), and *Normalized Variation*

* Corresponding author.

Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 158 - 169

of Branching Entropy (Magistry and Sagot, 2012). The others, however, rely on generative statistical models whose parameters can be estimated from the target texts only, including *Hierarchical Dirichlet Process* (Goldwater et al., 2009), *Nested Pitman-Yor Process* (Mochihashi et al., 2009), *Bayesian HMM* (Chen et al., 2014), TopWORDS (Deng et al., 2016) and GTS (Yuan et al., 2020).

In general, methods based on word matching and unsupervised learning cannot produce high-quality text segmentation (Zhao and Kit, 2011), although some unsupervised methods are successful on word discovery (Deng et al., 2016). Methods based on supervised learning can achieve excellent performance in closed test (Emerson, 2005), but often suffer from dramatic performance degradation when applied to open-domain Chinese corpus containing many unregistered words and named entities (Liu and Zhang, 2012; Wang et al., 2019). Methods based on deep learning are usually more robust under the "pre-training and fine-tuning" framework, but still suffer from unstable performance and often fail to correctly segment technical words, which play a key role in deciphering the meaning of domain-specific texts, when applied to open-domain texts (Zhao et al., 2018; Fu et al., 2020). There are also some efforts in the literature to integrate supervised and unsupervised methods for improved performance (Zhao and Kit, 2007, 2008, 2011; Wang et al., 2019; Yang et al., 2019). But, these methods either heavily depend on manually labelled corpus for model training, or suffer from unbalanced emphasis on text segmentation and word discovery, resulting in limited improvement for CNLP in open domain. These facts make processing open-domain Chinese texts a critical bottleneck in computational linguistics even for today.

Many factors contribute to the stagnation on development of efficient tools for processing open-domain Chinese texts. From the methodology point of view, we do not have a proper learning framework yet to connect the text segmentation problem to the word discovery problem and deal with them at the same time effectively. From the practical point of view, the lack of proper evaluation criterion in open domain places a critical barrier for fair comparison of different methods and discourages researchers from looking for potential solutions.

This study tries to provide solutions to these critical issues. First, we propose a novel Bayesian framework to integrate TopWORDS, an effective word discoverer (Deng et al., 2016), and PKUSEG, a strong text segmenter, leading to a more efficient text segmenter called TopWORDS-Seg, which can achieve effective text segmentation and word discovery simultaneously in open domain. Next, we design a cocktail strategy for method evaluation and comparison by measuring the overall performance of a target method on both text segmentation in benchmark corpus and technical word discovery and segmentation in open-domain corpus. Experimental studies demonstrate that the proposed TopWORDS-Seg outperforms existing methods with a significant margin for CNLP in open domain.

2 TopWORDS-Seg

Proposed by Deng et al. (2016), TopWORDS is a general approach for offline natural language processing based on unsupervised statistical learning. Assuming that sentences are generated by randomly sampling and concatenating words from an underlying word dictionary (i.e., unigram language model), TopWORDS starts with an over-complete initial word dictionary $\mathcal{D}$ containing all plausible word candidates in the target texts, and gradually simplifies the model by removing non-significant word candidates from $\mathcal{D}$ based on statistical model selection principles, with the unknown word usage frequencies estimated by EM algorithm (Dempster et al., 1977).

TopWORDS is closely related to methods widely used in neural machine translation for constructing sub-word dictionary, and can be viewed as an advanced version of *WordPiece* (Schuster and Nakajima, 2012), *Byte Pair Encoding* (Sennrich et al., 2016) and *Unigram Language Model* (Kudo, 2018). In practice, TopWORDS is particularly effective on discovering words, technical terms and phrases from open-domain Chinese texts, but tends to segment texts with coarser granularity at phrase instead of word level.

In this section, we upgrade TopWORDS from a weak text segmenter with strong ability on word discovery to a more powerful tool enjoying balanced ability on both dimensions via Bayesian inference.

2.1 The Bayesian Framework

Following the setting in Deng et al. (2016), let $\mathcal{T} = \{T_1, \cdots, T_n\}$ be a collection of unsegmented Chinese text sequences to process, $\mathcal{A} = \{a_1, a_2, \cdots, a_M\}$ be the set of Chinese characters

involved in $\mathcal{T}$, and $\mathcal{D}_{\mathcal{T}}$ be the underlying vocabulary behind $\mathcal{T}$ unknown to the investigator. We aim to discover $\mathcal{D}_{\mathcal{T}}$ from $\mathcal{T}$, and predict the invisible *word boundary profile* $B_j = (b_{j1}, \cdots, b_{jL_j})$ for each piece of unsegmented Chinese text $T_j = a_{j1}a_{j2}\cdots a_{jL_j}e$, where $b_{jl} = 1$ if there is a word boundary behind the l-th position of T_j and 0 otherwise, and e is a special end mark indicating the end of text sequence.

To learn $\mathcal{D}_{\mathcal{T}}$, we starts with an over-complete initial word dictionary $\mathcal{D} = \{w_1, w_2, \ldots, w_N, e\}$ covering all plausible word candidates in $\mathcal{T}$ (i.e., all sub-strings in $\mathcal{T}$ whose length $\leq \tau_L$ and frequency $\geq \tau_F$) and the end mark e. For simplicity, we always assume that $\mathcal{D}_{\mathcal{T}} \subset \mathcal{D}$ and all characters in $\mathcal{A}$ are covered by $\mathcal{D}$.

Under the unigram language model, we have the following likelihood function for a piece of unsegmented text T_j given B_j and $\mathcal{D}$:

$$\mathbb{P}(T_j \mid \mathcal{D}, \boldsymbol{\theta}, B_j) = \prod_{w \in \mathcal{D}} (\theta_w)^{n_w(B_j)}, \quad (1)$$

where $\boldsymbol{\theta} = \{\theta_w\}_{w \in \mathcal{D}}$ with θ_w being the usage frequency of word w in $\mathcal{T}$, and $n_w(B_j)$ counts the number of occurrences of word w in the segmented version of T_j based on B_j. Let $\boldsymbol{B} = \{B_1, \cdots, B_n\}$ being the word boundary profiles of the n text sequences in $\mathcal{T}$. We have

$$\begin{aligned} \mathbb{P}(\mathcal{T} \mid \mathcal{D}, \boldsymbol{\theta}, \boldsymbol{B}) &= \prod_{j=1}^{n} \mathbb{P}(T_j \mid \mathcal{D}, \boldsymbol{\theta}, B_j) \\ &= \prod_{w \in \mathcal{D}} (\theta_w)^{n_w(\boldsymbol{B})}, \quad (2) \end{aligned}$$

where

$$n_w(\boldsymbol{B}) = \sum_{j=1}^{n} n_w(B_j).$$

In this study, we propose to specify a joint prior distribution $\pi(\boldsymbol{\theta}, \boldsymbol{B})$ for $(\boldsymbol{\theta}, \boldsymbol{B})$ to integrate prior preference on word usage and text segmentation into the learning procedure. According to the Bayes Theorem, we have the following posterior distribution of $(\boldsymbol{\theta}, \boldsymbol{B})$ given $\mathcal{T}$ and $\mathcal{D}$:

$$\mathbb{P}(\boldsymbol{\theta}, \boldsymbol{B} \mid \mathcal{T}, \mathcal{D}) \propto \pi(\boldsymbol{\theta}, \boldsymbol{B}) \cdot \mathbb{P}(\mathcal{T} \mid \mathcal{D}, \boldsymbol{\theta}, \boldsymbol{B}),$$

which leads to the following marginal and conditional posterior distributions:

$$\begin{aligned} \mathbb{P}(\boldsymbol{\theta} \mid \mathcal{T}, \mathcal{D}) &= \int \mathbb{P}(\boldsymbol{\theta}, \boldsymbol{B} \mid \mathcal{T}, \mathcal{D}) d\boldsymbol{B}, \\ \mathbb{P}(\boldsymbol{B} \mid \mathcal{T}, \mathcal{D}, \boldsymbol{\theta}) &\propto \mathbb{P}(\boldsymbol{\theta}, \boldsymbol{B} \mid \mathcal{T}, \mathcal{D}). \end{aligned}$$

Based on $\mathbb{P}(\boldsymbol{\theta} \mid \mathcal{T}, \mathcal{D})$, model parameters $\boldsymbol{\theta}$ can be estimated by the posterior mode, i.e,

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \mathbb{P}(\boldsymbol{\theta} \mid \mathcal{T}, \mathcal{D}). \quad (3)$$

Given $\hat{\boldsymbol{\theta}}$, we can further infer $\boldsymbol{B}$ according to $\mathbb{P}(\boldsymbol{B} \mid \mathcal{T}, \mathcal{D}, \hat{\boldsymbol{\theta}})$ to achieve text segmentation.

2.2 Specification of Prior Distribution

There are various ways to specify the prior distributions $\pi(\boldsymbol{\theta}, \boldsymbol{B})$. In this study, we choose to use the independent conjugate prior below for conceptual and computational convenience:

$$\pi(\boldsymbol{\theta}, \boldsymbol{B}) = \pi(\boldsymbol{\theta}) \cdot \pi(\boldsymbol{B}),$$

where

$$\begin{aligned} \pi(\boldsymbol{\theta}) &= Dirichlet(\boldsymbol{\theta} \mid \boldsymbol{\alpha}), \\ \pi(\boldsymbol{B}) &= \prod_{j=1}^{n} \pi(B_j) = \prod_{j=1}^{n} \prod_{l=1}^{L_j} \pi(b_{jl}), \\ \pi(b_{jl}) &= Binary(b_{jl} \mid \rho_{jl}), \end{aligned}$$

with $\boldsymbol{\alpha} = \{\alpha_w\}_{w \in \mathcal{D}}$ and $\boldsymbol{\rho} = \{\rho_{jl}\}$ being the hyper-parameters controlling the strength of prior information.

In this study, we choose to specify

$$\alpha_w = 1, \ \forall \ w \in \mathcal{D}, \quad (4)$$

leading to a flat prior distribution for $\boldsymbol{\theta}$, but adopt a non-flat prior distribution for $\boldsymbol{\rho}$ by smoothing the word boundary profiles $\boldsymbol{B}^* = \{B_j^*\}_{1 \leq j \leq n}$ predicted by a pre-given text segmenter $\mathcal{S}^*$:

$$\rho_{jl} = \begin{cases} (1-\kappa) \cdot b_{jl}^* + \kappa \cdot \rho, & l < L_j, \\ 1, & l = L_j, \end{cases} \quad (5)$$

where b_{jl}^* is the location-specific binary segmentation indicator predicted by $\mathcal{S}^*$, $\kappa \in (0, 1)$ is the smoothing parameter, and $\rho > 0$ highlights the probability to place a word boundary at each location by a pseudo segmenter that places boundaries randomly in the text sequence.

Here, we set $\rho = 0.5$ by default, and leave κ as a hyper-parameter that can be tuned to fit different application scenarios, leading to the following joint prior distribution:

$$\pi_{\kappa}(\boldsymbol{\theta}, \boldsymbol{B}) \propto \prod_{j=1}^{n} \prod_{l=1}^{L_j} (\rho_{jl})^{b_{jl}} (1-\rho_{jl})^{1-b_{jl}}. \quad (6)$$

2.3 Word Discovery

Given the prior distribution $\pi_\kappa(\boldsymbol{\theta}, \boldsymbol{B})$ specified previously, the posterior distribution becomes:

$$\begin{aligned} & \mathbb{P}(\boldsymbol{\theta}, \boldsymbol{B} \mid \mathcal{T}, \mathcal{D}) \\ \propto\ & \pi_\kappa(\boldsymbol{\theta}, \boldsymbol{B}) \cdot \mathbb{P}(\mathcal{T} \mid \mathcal{D}, \boldsymbol{\theta}, \boldsymbol{B}) \\ \propto\ & \prod_{j=1}^{n} \left[\pi_\kappa(B_j) \cdot \prod_{w \in \mathcal{D}} (\theta_w)^{n_w(B_j)} \right], \end{aligned} \tag{7}$$

where

$$\pi_\kappa(B_j) = \prod_{l=1}^{L_j} (\rho_{jl})^{b_{jl}} (1 - \rho_{jl})^{1 - b_{jl}}$$

is a deterministic function of κ, as ρ_{jl}'s degenerate to constants for fixed κ based on (5). Under such a Bayesian model, the problem of word discovery can be naturally converted into a statistical model selection problem, as only word candidates whose usage frequency θ_w is significantly larger than 0 could be meaningful words. We estimate $\boldsymbol{\theta}$ by the posterior mode $\hat{\boldsymbol{\theta}}$ as defined in (3), which can by obtained via the EM algorithm (Dempster et al., 1977) with $\boldsymbol{B}$ as the missing data. Details of the EM algorithm are described in Appendix A.

Once the EM algorithm gets converged, we can evaluate the statistical significance of a word candidate w by the likelihood-ratio statistics between the full model and a reduced model with w removed:

$$\psi_w = \log \left(\frac{\mathbb{P}(\mathcal{T} \mid \mathcal{D}, \hat{\boldsymbol{\theta}})}{\mathbb{P}(\mathcal{T} \mid \mathcal{D}, \hat{\boldsymbol{\theta}}_{[w=0]})} \right), \tag{8}$$

where $\hat{\boldsymbol{\theta}}_{[w=0]}$ is the modification of $\hat{\boldsymbol{\theta}}$ by setting $\hat{\theta}_w = 0$ with other elements unchanged. Apparently, a larger ψ_w suggests that word candidate w is more important for fitting the observed texts, and thus is more likely to be a meaningful word. Because $-2\psi_w \sim \chi^2$ asymptotically under the null hypothesis that the reduced model with w removed is the true model, we can filter out word candidates whose $\psi_w < \tau_\psi$, where threshold τ_ψ is the $(1 - \frac{0.05}{N})$-quantile of the χ^2 distribution, following the Bonferroni correction principle for multiple hypothesis testing. As demonstrated by Deng et al. (2016), such a model selection strategy can effectively filter out most meaningless word candidates and results in a concise final dictionary containing meaningful words and phrases only.

Considering that

$$\psi_w = -\sum_{j=1}^{n} \log\left(1 - r_{wj}\right),$$

where

$$\begin{aligned} r_{wj} &= \mathbb{P}_\kappa\left(w \sim B_j \mid T_j, \mathcal{D}, \hat{\boldsymbol{\theta}}\right) \\ &= \sum_{B_j \in \mathcal{B}_j} I\left(w \sim B_j\right) \cdot \mathbb{P}_\kappa(B_j \mid T_j, \mathcal{D}, \hat{\boldsymbol{\theta}}), \end{aligned} \tag{9}$$

with notation "$w \sim B_j$" meaning that word candidate w appears in the segmented version of T_j based on B_j, we can get ψ_w by calculating r_{wj} for each T_j.

2.4 Text Segmentation

Given $\hat{\boldsymbol{\theta}}$, plausible text segmentation of T_j can be obtained by optimizing B_j according to $\mathbb{P}_\kappa(B_j \mid T_j, \mathcal{D}, \hat{\boldsymbol{\theta}})$, i.e., segment T_j according to

$$\hat{B}_j = \max_{B \in \mathcal{B}_j} \mathbb{P}_\kappa(B \mid T_j, \mathcal{D}, \hat{\boldsymbol{\theta}}). \tag{10}$$

Alternatively, we can also calculate the posterior probability of existing a word boundary at position (j, l) as

$$\gamma_{jl} = \sum_{B \in \mathcal{B}_j} b_{jl} \cdot \mathbb{P}_\kappa(B_j \mid T_j, \mathcal{D}, \hat{\boldsymbol{\theta}}), \tag{11}$$

and segment T_j based on

$$\tilde{B}_j = I(\boldsymbol{\gamma}_j \geq \tau_S), \tag{12}$$

where $\boldsymbol{\gamma}_j = (\gamma_{j1}, \cdots, \gamma_{jL_j})$ and τ_S is a pre-given threshold with 0.5 as the default value. Here, we choose to use the second segmentation strategy, because it leads to more robust results in practice.

2.5 TopWORDS-Seg Algorithm

Integrating the dictionary initialization stage via sub-string enumeration, the prior construction stage guided by a pre-given segmenter $\mathcal{S}^*$ (i.e., PKUSEG by default), the word discovery stage empowered by EM algorithm and likelihood-ratio tests, and the text segmentation stage based on conditional probability inference, into a united framework, we come up with the TopWORDS-Seg algorithm as demonstrated in Figure 1. Computation issues involved in the algorithm are detailed in Appendix B.

A collection of hyper-parameters, including τ_L, τ_F, κ, ρ and τ_S, are associated with the TopWORDS-Seg algorithm, and need be specified to initiate the algorithm. We recommend to set $\tau_L = 15$, $\tau_F = 2$ and $\rho = \tau_S = 0.5$ by default. The specification of hyper-parameter κ is a bit complicated. To capture unregistered words from open-domain texts more efficiently, we would like to

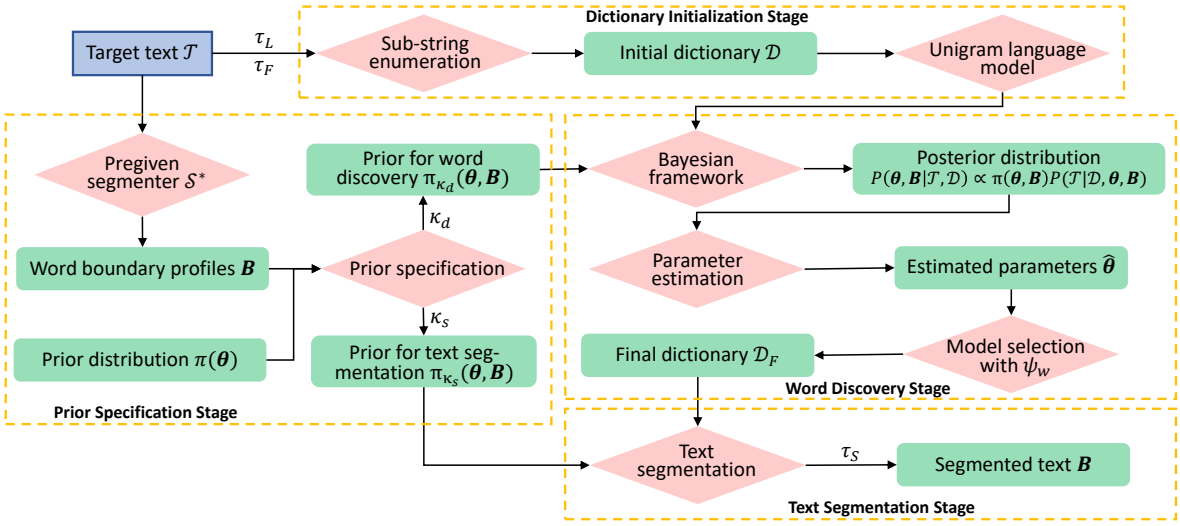


Figure 1: Flow chart of the TopWORDS-Seg

choose a larger κ to encourage word discovery. To segment regular texts more precisely, however, we would like to choose a smaller κ instead to better utilize the prior information. To get rid of the dilemma, we allow to specify κ with different values in different tasks, i.e., using a large κ (referred to as κ_d) in the word discovery stage and a small κ (referred to as κ_s) in the text segmentation stage. Based on a wide range of experimental studies, we suggest to set $\kappa_d = 0.5$ and $\kappa_s = 0.001$ by default.

3 Experimental Study on Wikipedia

Composed of over 10 billion Chinese character tokens from 3.6 million webpages, Chinese Wikipedia (https://dumps.wikimedia.org/) is one of the largest open-source Chinese corpus. Containing rich contents of various domains and millions of technical terms highlighted by hyperlinks, the Chinese Wikipedia is an ideal corpus for studying CNLP in open domain.

Considering that it's computationally expensive to processing all webpages in Chinese Wikipedia, we randomly picked up 1,500 webpages involving 8 million Chinese character tokens (referred to as Chinese Wiki-Rand, or $\mathcal{T}_{W\text{-}R}$) as the representative samples of the general texts in Chinese Wikipedia. Moreover, we selected two collections of special webpages from Chinese Wikipedia with label "电影" (referred to as Chinese Wiki-Film, or $\mathcal{T}_{W\text{-}F}$) or "物理" (referred to as Chinese Wiki-Physics, or $\mathcal{T}_{W\text{-}P}$), involving ~5 million Chinese character tokens for each, as the representatives of the domain-specific texts in Chinese Wikipedia. Figure 2 (a) and (b) demonstrates a typical Wikipedia web page and histograms for term length and appearance frequency of technical terms involved in $\mathcal{T}_{W\text{-}R}$.

In this section, we apply TopWORDS-Seg to process these Wikipedia corpora separately, and compare its performance to 6 existing methods, including Jieba (Sun, 2012), StanfordNLP (Manning et al., 2014), THULAC (Sun et al., 2016), PKUSEG (Luo et al., 2019), LTP (Che et al., 2021), and TopWORDS (Deng et al., 2016) itself, from various aspects.

3.1 Performance Evaluation Criteria

Due to the lack of gold standard, it is not straightforward to evaluate and compare the performance of different methods on open-domain corpus like Chinese Wikipedia. Here, we propose a cocktail strategy for method evaluation by measuring the overall performance of each method on both open-domain corpuora and benchmark corpus.

Let V_t be the collection of frequent technical terms in a particular Wikipedia corpus (terms with hyperlinks appear at least 2 times), with n_w be the number of occurrences for each $w \in V_t$. Suppose V is the discovered vocabulary reported by a particular method $\mathcal{M}$, and m_w is the number of successful catches of w by $\mathcal{M}$. Taking advantage of the self-labelled technical terms with hyperlinks in Wikipedia webpages, it is straightforward to measure *discovery recall* R_d and *segmentation recall* R_s for technical terms in V_t as below:

$$R_d = \frac{|V_t \cap V|}{|V_t|} \quad \text{and} \quad R_s = \frac{\sum_{w \in V_t} m_w}{\sum_{w \in V_t} n_w}. \quad (13)$$

Together, R_d and R_s reflect the ability of method

$\mathcal{M}$ to deal with technical terms in open-domain texts.

Because it is difficult to directly evaluate the perform of a method $\mathcal{M}$ on segmenting non-technical contents of the Wikipedia corpus, we retreat to indirect evaluation by evaluating its performance on segmenting the PKU corpus $\mathcal{T}_P$, a benchmark corpus with gold standard released by SIGHAN 2005 Bake-Off (Emerson, 2005), instead. Let F_s be the F_1 score of method $\mathcal{M}$ on text segmentation for the PKU corpus. Score F_s reflects $\mathcal{M}$'s ability to process general Chinese texts without technical contents.

Apparently, R_d, R_s and F_s measure the strength of a method comprehensively from various aspects, with both word discovery and text segmentation considered for technical as well as non-technical texts. Such a cocktail strategy provide us a principle to evaluate and compare the overall performance of different CNLP methods in open domains. If a method enjoys high R_d, R_s and F_s values across different corpora stably, we would feel comfortable to claim it as a robust tools for CNLP in open domains.

3.2 Results

Figure 2 (c) summarizes the performance of TopWORDS-Seg (with the default setting) and the 6 competing methods on the Wikipedia and PKU corpora in terms of R_d, R_s and F_s, with the size of discovered vocabulary $|V|$ reported as well. Comparing these results, we find that TopWORDS-Seg enjoys robust performance on segmenting classic benchmark corpus ($F_s = 82.2\%$ for $\mathcal{T}_P$), open-domain corpus ($R_s = 76.5\%$ for $\mathcal{T}_{W\text{-}R}$) and domain-specific corpus ($R_s = 76.8\%$ and 70.8% for $\mathcal{T}_{W\text{-}F}$ and $\mathcal{T}_{W\text{-}P}$ respectively), and high efficiency on discovering technical terms ($R_d > 82\%$ for all three Wikipedia corpora). The other methods, however, all suffer from either missing too many technical terms in the Wikipedia corpora (R_d ranging from 45% to 77% as in supervised methods), or segmenting the PKU corpus poorly ($F_s = 50.4\%$ as in TopWORDS). Considering that TopWORDS-Seg reports a vocabulary that is 16K smaller than TopWORDS, it actually outperforms TopWORDS significantly in all dimensions.

Moreover, considering that both TopWORDS and TopWORDS-Seg tend to segment Chinese texts at coarser granularity with technical terms and phrases preserved as composite words instead of cutting them into smaller language units, the text segmentation standard adopted by the PKU corpus, which tends to segment Chinese texts at finer granularity, may over-punish them. To ease the impact on performance evaluation due to segmentation granularity, we choose to mask part of the PKU corpus $\mathcal{T}_P$ where method $\mathcal{M}$ is not consistent with the standard segmentation only on granularity (with the concrete criteria detailed in Appendix C), and measure the F_1 score of method $\mathcal{M}$ on the masked version of $\mathcal{T}_P$ only, leading to a masked version of F_s referred to as F_m. The proportion of masked corpus (i.e., *mask rate*) is also calculated for each method and reported in Figure 2 (c). TopwORDS-Seg achieves an improved $F_m = 93.7\%$ with a mask rate of 16.6%, suggesting that TopwORDS-Seg actually segments the PKU corpus very well. Meanwhile, a much higher mask rate of 50.4% is obtained for TopWORDS, which is consistent to our impression that TopWORDS tends to preserve too many sub-phrases in text segmentation.

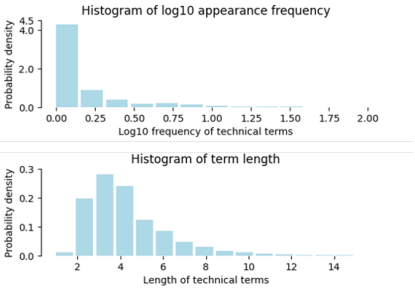
In addition, because some methods based on supervised learning, e.g., Jieba, THULAC and PKUSEG, can receive external vocabulary for processing open-domain corpus, there exists an alternative strategy to integrate TopWORDS with thses methods by simply forwarding the vocabulary discovered by TopWORDS to them. We refer to approaches based on this strategy as TopWORDS-Jieba/THULAC/PKUSEG, and report their performance on both Chinese Wikipedia corpus and PKU corpus in Figure 2 (c) as well. Unfortunately, although this family of approaches achieve a higher R_d in general, they tend to report an over-large vocabulary and segment texts with coarser granularity like TopWORDS does. These results indicate that simply concatenating TopWORDS to other methods does not necessarily lead to an improved approach, and thus imply that the proposed strategy based on Bayesian inference is not trivial.

The heatmaps in Figure 2 (d) demonstrate the similarity on text segmentation of different methods on four different target corpora, where the similarity between any two methods $\mathcal{M}_i$ and $\mathcal{M}_j$ is measured by

$$\phi_{ij} = \frac{\sum_{T\in\mathcal{T}_D} sum(B_T^{(i)} \wedge B_T^{(j)})}{\sum_{T\in\mathcal{T}_D} sum(B_T^{(i)} \vee B_T^{(j)})},$$

with $B_T^{(i)}$ denoting the predicted word boundary vector of text sequence T by method $\mathcal{M}_i$. From the figure, we can see clearly that text segmentation

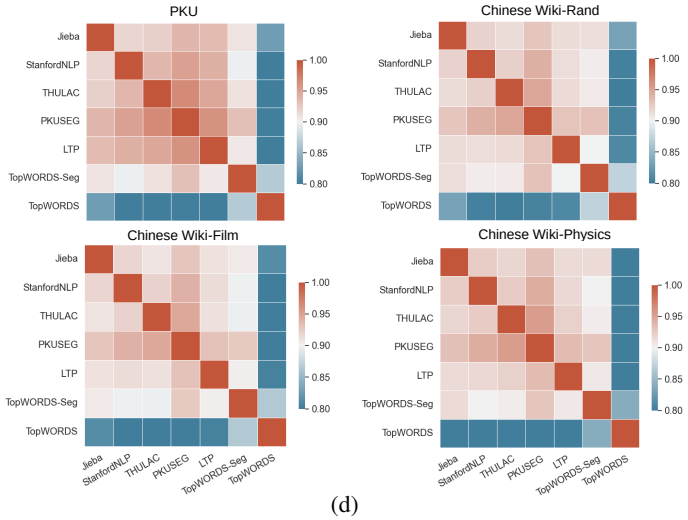
(a)



(b)

Method	ChineseWiki-Rand			ChineseWiki-Film			ChineseWiki-Physics			PKU		
	$\|V\|$	R_d	R_s	$\|V\|$	R_d	R_s	$\|V\|$	R_d	R_s	F_s	F_m	Mask Rate
Jieba	110k	60.2%	72.6%	67k	48.6%	60.0%	43k	47.0%	59.9%	81.2%	98.6%	22.4%
StanfordNLP	100K	58.1%	64.1%	64k	47.7%	55.3%	43k	45.6%	49.1%	85.8%	93.9%	11.4%
THULAC	101K	59.4%	64.8%	60k	46.4%	52.6%	42k	47.1%	49.2%	92.4%	95.6%	4.5%
PKUSEG	105k	56.9%	63.8%	63k	46.2%	53.7%	43k	45.3%	49.9%	95.4%	99.5%	5.5%
LTP	130k	76.4%	67.2%	78k	65.1%	63.5%	63k	72.4%	59.3%	88.7%	99.8%	14.7%
TopWORDS	165k	86.8%	71.9%	103k	82.5%	72.7%	92k	85.7%	61.6%	50.4%	85.8%	50.4%
TopWORDS-Seg	149K	86.9%	76.5%	92k	82.0%	76.8%	80k	85.1%	70.8%	82.2%	93.7%	16.6%
TopWORDS-Jieba	201K	91.4%	72.8%	120k	85.1%	73.1%	104k	89.1%	60.9%	50.9%	95.8%	55.0%
TopWORDS-THULAC	193K	91.8%	71.8%	116k	85.2%	73.2%	103k	89.5%	61.5%	54.9%	98.4%	52.6%
TopWORDS-PKUSEG	214K	90.0%	69.5%	127k	85.0%	71.4%	117k	89.1%	57.7%	44.5%	77.2%	51.3%

(c)



(d)

Target text: 碳的各种同素异形体的物理特性差异极大

(The physical properties of various allotropes of carbon are extremely different)

Method	Segmented text
PKUSEG	碳|的|各种|同|素异|形体|的|物理|特性|差异|极|大
TopWORDS	碳的|各种|同素异形体|的|物理特性|差异极大
TopWORDS-Seg	碳|的|各种|同素异形体|的|物理|特性|差异|极|大

(e)

Figure 2: Experimental study on PKU corpus and 3 Chinese Wikipedia corpora. (a) A typical web page in Chinese Wikipedia. (b) Key characteristics of technical terms involved in Chines Wikipedia. (c) Results on PKU, Chinese Wiki-Rand, Chinese Wiki-Film and Chinese Wiki-Physics datasets of different methods. (d) Similarity on text segmentation of different methods on four different target corpora. (e) Segmentation results on a typical sentence

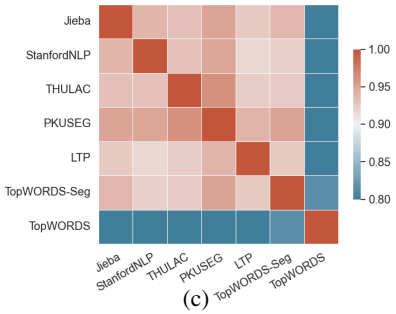
reported by TopWORDS-Seg is very similar to the results reported by supervised methods, but is significantly different from the result reported by TopWORDS for all four corpora. Such results confirm the strength of TopWORDS-Seg on text segmentation in addition to word discovery, and provide strong evidences to support TopWORDS-Seg as a powerful tool for processing open-domain Chinese texts.

Figure 2 (e) shows an illustrative example of text segmentation of PKUSEG, TopWORDS and TopWORDS-Seg for a piece of target text, respectively. Apparently, PKUSEG segments the target text almost perfectly except for chopping the technical term *allotropes* (同素异形体) into three substrings by mistake, due to the lack of ability to recognize unregistered words. TopWORDS, however, successfully recognizes and segments the technical term *allotropes* correctly, but segments the other part of the target text with coarser granularity leaving phrases like *physical properties* (物理特性) and *extremely different* (差异极大) as unsegmented language units. TopWORDS-Seg, as expected, segments the target text perfectly, with

(a)

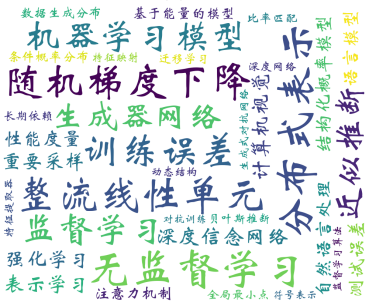
Method	$\|V\|$	R_d
Jieba	7.0K	32.1%
StanfordNLP	7.0K	36.0%
THULAC	6.8K	35.4%
PKUSEG	6.8K	36.0%
LTP	12.2K	69.2%
TopWORDS	12.8K	85.0%
TopWORDS-Seg	10.7K	84.1%

(b)



(c)

(d)



(e)

(f)

Figure 3: Real application on the full text of the Chinese version of *Deep Learning*. (a) Cover page of the book. (b) Performance on word discovery of different methods. (c) Similarity on text segmentation of different methods. (d) 100 most frequent words discovered by TopWORDS-Seg. (e) Technical terms captured by TopWORDS-Seg but missed by all supervised methods. (f) Typical pseudo words and phrases reported by TopWORDS but eliminated by TopWORDS-Seg.

the technical term *allotropes* correctly recognized and the rest part segmented with proper granularity.

4 Processing the Book of *Deep Learning*

Written by Goodfellow et al. (2016), the book *Deep Learning* has become a classic tutorial for deep learning. In 2017, its Chinese version was published in China (see Figure 3 (a) for the book's cover), which is composed of more than 400,000 Chinese character tokens (referred to as $\mathcal{T}_D$). Covering rich technical contents in the domain of machine learning, including over 800 technical terms as listed in the Index Table at the end of the book, such a book is an ideal target for testing the performance of the proposed TopWORDS-Seg in real application.

Feeding full text of the book to TopWORDS-Seg and competing methods respectively, we obtained results as summarized in Figure 3. Figure 3 (b) shows that TopWORDS-Seg discovers 84.1% technical terms listed in the Index Table of the book with a vocabulary of 10.7K discovered words. TopWORDS achieves a slightly higher $R_d = 85.0\%$ at the price of a larger vocabulary with 12.8K discovered words. Other methods based on supervised learning result in much lower R_d with the vocabulary size varying between 6.8K to 12.2K. Figure 3 (d) shows the most frequent words discovered by TopWORDS-Seg. Figure 3 (e) displays part of the technical terms captured by TopWORDS-Seg but missed by all supervised methods, which are all meaningful technical terms like *unsupervised learning* (无监督学习) and *stochastic gradient decent* (随机梯度下降). Figure 3 (f) summarizes typical pseudo words and phrases reported by TopWORDS but eliminated by TopWORDS-Seg, which are all common collocations widely used but usually not treated as words in Chinese, e.g., *in the model* (模型中) and *it is because of* (是因为). These results suggest that TopWORDS-Seg is indeed more effective than competing methods on word discovery.

In terms of text segmentation, the heatmap in Figure 3 (c) visualizes the similarity between TopWORDS-Seg and other approaches on this corpus in a similar fashion as in Figure 2 (d). Again, the performance of TopWORDS-Seg is very similar to the supervised methods, and demonstrates significant difference from TopWORDS, suggesting that TopWORDS-Seg is a robust tool with balanced ability on processing open-domain Chinese texts.

5 Conclusions and Discussions

In this paper, we proposed TopWORDS-Seg, a powerful tool for processing open-domain Chi-

nese texts based on Bayesian inference with balanced ability on text segmentation and word discovery. A series of experimental studies confirm that TopWORDS-Seg can discover unregistered technical terms in open-domain texts effectively, and achieve high-quality text segmentation on both benchmark and open-domain corpora. Taking advantage of the Bayesian framework, TopWORDS-Seg is ready to process large scale open-domain Chinese texts without extra training corpus or pregiven domain vocabulary, leading to an ideal solution to a critical bottleneck existing in computational linguistics for decades. Moreover, combing the strong points of PKUSEG and TopWORDS via Bayesian inference, TopWORDS-Seg enjoys transparent reasoning process, and is fully interpretable to most people. In practical applications, such a property is very attractive to many researchers and practicers.

Meanwhile, TopWORDS-Seg also suffers from a few obvious limitations. For example, although the current learning framework is effective to discover frequent words, it tends to miss many rare words that appear only a few times in the texts. For another instance, because PKUSEG is more reliable on segmenting general texts, but less reliable on segmenting technical texts, in the ideal case we should adopt prior information provided by PKUSEG adaptively when processing texts of different types. Unfortunately, TopWORDS-Seg does not take such a natural idea into consideration yet, and simply use the PKUSEG prior at the same intensity everywhere. These deficiencies partially explain why TopWORDS-Seg still misses about 15% technical terms in both experimental studies reported in this paper. More research efforts are needed to fill in these gaps in future.

Acknowledgements

This research is partially supported by the National Scientific and Technological Innovation 2030 Major Project (No: 2020AAA0106501), the Guo Qiang Institute of Tsinghua University, the Beijing Natural Science Foundation (Z190021), and the Scientific-Technological Innovation Plan Program of Universities guided by the Ministry of Education of China. Changzai Pan is supported by China Scholarship Council.

References

Adam Berger, Stephen A Della Pietra, and Vincent J Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational linguistics*, 22(1):39–71.

Jason S. Chang and Tracy Lin. 2003. Unsupervised word segmentation without dictionary. In *ROCLING 2003 Poster Papers*, pages 355–359, Hsinchu, Taiwan. The Association for Computational Linguistics and Chinese Language Processing (ACLCLP).

Wanxiang Che, Yunlong Feng, Libo Qin, and Ting Liu. 2021. N-LTP: An open-source neural language technology platform for Chinese. *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 42–49.

Aitao Chen. 2003. Chinese word segmentation using minimal linguistic knowledge. In *Proceedings of the Second SIGHAN Workshop on Chinese Language Processing*, pages 148–151, Sapporo, Japan. Association for Computational Linguistics.

Keh-Jiann Chen and Shing-Huan Liu. 1992. Word identification for mandarin Chinese sentences. In *Proceedings of the 14th Conference on Computational Linguistics - Volume 1*, COLING '92, page 101–107, USA. Association for Computational Linguistics.

Miaohong Chen, Baobao Chang, and Wenzhe Pei. 2014. A joint model for unsupervised Chinese word segmentation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 854–863, Doha, Qatar. Association for Computational Linguistics.

Xinchi Chen, Xipeng Qiu, Chenxi Zhu, Pengfei Liu, and Xuanjing Huang. 2015. Long short-term memory neural networks for Chinese word segmentation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1197–1206, Lisbon, Portugal. Association for Computational Linguistics.

Arthur P Dempster, Nan M Laird, and Donald B Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22.

Ke Deng, Peter K. Bol, Kate J. Li, and Jun S. Liu. 2016. On the unsupervised analysis of domain-specific Chinese texts. *Proceedings of the National Academy of Sciences*, 113(22):6154–6159.

Thomas Emerson. 2005. The second international Chinese word segmentation bakeoff. In *Proceedings of the fourth SIGHAN workshop on Chinese language Processing*.

Haodi Feng, Kang Chen, Xiaotie Deng, and Weimin Zheng. 2004. Accessor variety criteria for Chinese word extraction. *Computational Linguistics*, 30(1):75–93.

Jinlan Fu, Pengfei Liu, Qi Zhang, and Xuanjing Huang. 2020. RethinkCWS: Is Chinese word segmentation a solved task? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5676–5686, Online. Association for Computational Linguistics.

P. Geutner. 1996. Introducing linguistic constraints into statistical language modeling. In *Proceeding of Fourth International Conference on Spoken Language Processing. ICSLP '96*, volume 1, pages 402–405.

Sharon Goldwater, Thomas L. Griffiths, and Mark Johnson. 2009. A Bayesian framework for word segmentation: Exploring the effects of context. *Cognition*, 112(1):21–54.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press. `http://www.deeplearningbook.org`.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF models for sequence tagging. *arXiv preprint arXiv:1508.01991*.

Chunyu Kit and Yorick Wilks. 1999. Unsupervised learning of word boundary with description length gain. In *EACL 1999: CoNLL-99 Computational Natural Language Learning*.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia. Association for Computational Linguistics.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, page 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Yang Liu and Yue Zhang. 2012. Unsupervised domain adaptation for joint segmentation and POS-tagging. In *Proceedings of COLING 2012: Posters*, pages 745–754, Mumbai, India. The COLING 2012 Organizing Committee.

Jin Kiat Low, Hwee Tou Ng, and Wenyuan Guo. 2005. A maximum entropy approach to Chinese word segmentation. In *Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing*.

Ruixuan Luo, Jingjing Xu, Yi Zhang, Xuancheng Ren, and Xu Sun. 2019. PKUSEG: A toolkit for multi-domain Chinese word segmentation. *CoRR*, abs/1906.11455.

Ji Ma, Kuzman Ganchev, and David Weiss. 2018. State-of-the-art Chinese word segmentation with Bi-LSTMs. pages 4902–4908.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via Bi-directional LSTM-CNNs-CRF. pages 1064–1074.

Pierre Magistry and Benoît Sagot. 2012. Unsupervized word segmentation: the case for mandarin Chinese. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 383–387, Jeju Island, Korea. Association for Computational Linguistics.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland. Association for Computational Linguistics.

Andrew McCallum, Dayne Freitag, and Fernando CN Pereira. Maximum entropy markov models for information extraction and segmentation.

Daichi Mochihashi, Takeshi Yamada, and Naonori Ueda. 2009. Bayesian unsupervised word segmentation with nested Pitman-Yor language modeling. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 100–108, Suntec, Singapore. Association for Computational Linguistics.

Fuchun Peng, Fangfang Feng, and Andrew McCallum. 2004. Chinese segmentation and new word detection using conditional random fields. In *Proceedings of the 20th International Conference on Computational Linguistics*, COLING '04, page 562–568, USA. Association for Computational Linguistics.

Mike Schuster and Kaisuke Nakajima. 2012. Japanese and Korean voice search. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5149–5152.

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 (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Xinxin Shu, Junhui Wang, Xiaotong Shen, and Annie Qu. 2017. Word segmentation in Chinese language processing. *Statistics and Its Interface*, 10(2):165–173.

R. Sproat, Chilin Shih, W. Gale, and N. Chang. 1994. A stochastic finite-state word-segmentation algorithm for Chinese. *Proceedings of the 32nd Annual Meeting on Association for Computational Linguistics*, page 66–73.

Junyi Sun. 2012. Jieba Chinese word segmentation tool.

Maosong Sun, Xinxiong Chen, Kaixu Zhang, Zhipeng Guo, and Zhiyuan Liu. 2016. THULAC: An efficient lexical analyzer for Chinese.

Chunqi Wang and Bo Xu. Convolutional neural network with word embeddings for Chinese word segmentation. *arXiv preprint arXiv:1711.04411*.

Hanshi Wang, Jian Zhu, Shiping Tang, and Xiaozhong Fan. 2011. A new unsupervised approach to word segmentation. *Computational Linguistics*, 37(3):421–454.

Xiaobin Wang, Deng Cai, Linlin Li, Guangwei Xu, Hai Zhao, and Luo Si. 2019. Unsupervised learning helps supervised neural word segmentation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):7200–7207.

Nianwen Xue. 2003. Chinese word segmentation as character tagging. *International Journal of Computational Linguistics & Chinese Language Processing*, 8(1):29–48.

Haiqin Yang. 2019. BERT meets Chinese word segmentation. *arXiv preprint arXiv:1909.09292*. ArXiv: 1909.09292.

Yang Yang, Qi Li, Zhaoyang Liu, Fang Ye, and Ke Deng. 2019. Understanding traditional Chinese medicine via statistical learning of expert-specific electronic medical records. *Quantitative Biology*, 7(3):210–232.

Zheng Yuan, Yuanhao Liu, Qiuyang Yin, Boyao Li, Xiaobin Feng, Guoming Zhang, and Sheng Yu. 2020. Unsupervised multi-granular Chinese word segmentation and term discovery via graph partition. *Journal of Biomedical Informatics*, 110:103542.

Hua-Ping Zhang, Hong-Kui Yu, De-Yi Xiong, and Qun Liu. 2003. HHMM-based Chinese lexical analyzer ICTCLAS. In *Proceedings of the Second SIGHAN Workshop on Chinese Language Processing*, pages 184–187, Sapporo, Japan. Association for Computational Linguistics.

Hai Zhao and Chunyu Kit. 2007. Incorporating global information into supervised learning for Chinese word segmentation. In *10th Conference of the Pacific Association for Computational Linguistics*, pages 66–74.

Hai Zhao and Chunyu Kit. 2008. Exploiting unlabeled text with different unsupervised segmentation criteria for Chinese word segmentation. *Research in Computing Science*, 33:93–104.

Hai Zhao and Chunyu Kit. 2011. Integrating unsupervised and supervised word segmentation: The role of goodness measures. *Information Sciences*, 181(1):163–183.

Lujun Zhao, Qi Zhang, Peng Wang, and Xiaoyu Liu. 2018. Neural networks incorporating unlabeled and partially-labeled data for cross-domain Chinese word segmentation. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, page 4602–4608. AAAI Press.

A EM Algorithm for Estimating $\hat{\theta}$

Given $\boldsymbol{\theta}^{(t)}$, the current estimation of $\boldsymbol{\theta}$, the E-step computes the Q-function below:

$$\begin{aligned} & Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(t)}) \\ = \; & \mathbb{E}\left(\log\left(\mathbb{P}(\boldsymbol{\theta}, \boldsymbol{B} \mid \mathcal{T}, \mathcal{D})\right) \mid \mathcal{T}, \mathcal{D}, \boldsymbol{\theta}^{(t)}\right) \\ = \; & C + \sum_{w \in \mathcal{D}} \left(\log \theta_w \cdot n_w(\boldsymbol{\theta}^{(t)})\right), \end{aligned} \tag{14}$$

where C is constant that does not change with $\boldsymbol{\theta}$,

$$n_w(\boldsymbol{\theta}^{(t)}) = \sum_{j=1}^{n} n_{w,j}(\boldsymbol{\theta}^{(t)}), \tag{15}$$

$$\begin{aligned} n_{wj}(\boldsymbol{\theta}^{(t)}) &= \mathbb{E}\left(n_w(B_j) \mid T_j, \mathcal{D}, \boldsymbol{\theta}^{(t)}\right) \\ &= \sum_{B_j \in \mathcal{B}_j} n_w(B_j) \cdot \mathbb{P}_\kappa\left(B_j \mid T_j, \mathcal{D}, \boldsymbol{\theta}^{(t)}\right), \end{aligned} \tag{16}$$

$$\begin{aligned} & \mathbb{P}_\kappa\left(B_j \mid T_j, \mathcal{D}, \boldsymbol{\theta}^{(t)}\right) \\ = \; & \frac{\mathbb{P}(T_j \mid \mathcal{D}, \boldsymbol{\theta}^{(t)}, B_j) \cdot \pi_\kappa(B_j)}{\sum_{B \in \mathcal{B}_j} \mathbb{P}(T_j \mid \mathcal{D}, \boldsymbol{\theta}^{(t)}, B) \cdot \pi_\kappa(B)}, \end{aligned} \tag{17}$$

and $\mathcal{B}_j$ stands for the collection of all possible word boundary profiles of T_j. The M-step updates $\boldsymbol{\theta}^{(t)}$ by maximizing $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(t)})$ with respect to $\boldsymbol{\theta}$, leading to the updating function below:

$$\theta_w^{(t+1)} = \frac{n_w(\boldsymbol{\theta}^{(t)})}{\sum_{w \in \mathcal{D}} n_w(\boldsymbol{\theta}^{(t)})}, \; \forall \, w \in \mathcal{D}. \tag{18}$$

Along the updating procedure of the EM algorithm, word candidates with low estimated usage frequency (e.g., $\hat{\theta}_w < \tau_\theta = 10^{-8}$) can be gradually removed from $\mathcal{D}$ to simplify the model. When EM algorithm gets converged, we can get the estimation of posterior mode, $\hat{\boldsymbol{\theta}}$.

B Computational Details

Considering that

$$\psi_w = -\sum_{j=1}^{n} \log\left(1 - r_{wj}\right),$$

where

$$r_{wj} = \mathbb{P}_\kappa\left(w \sim B_j \mid T_j, \mathcal{D}, \hat{\boldsymbol{\theta}}\right) = \sum_{B_j \in \mathcal{B}_j} I\left(w \sim B_j\right) \cdot \mathbb{P}_\kappa(B_j \mid T_j, \mathcal{D}, \hat{\boldsymbol{\theta}}), \quad (19)$$

with notation "$w \sim B_j$" meaning that word candidate w appears in the segmented version of T_j based on B_j, we can get ψ_w by calculating r_{wj} for each T_j.

Thus, to implement the TopWORDS-Seg algorithm, we need to calculate n_{wj} in (15), r_{wj} in (19), $\hat{B}_j$ in (10) or γ_{jl} in (12) for $\forall\ T_j \in \mathcal{T}$. For a specific $T_j = T = a_1 \cdots a_L$, we define $T_{[t:s]} = a_t \cdots a_s$. It can be showed that n_{wj}, r_{wj} and γ_{jl}, which are all functions of T_j, have the formulation below:

$$n_w(T) = \frac{1}{p(T)} \sum_{1 \le t < s \le L} \Big[p(T_{[<t]}) \cdot p(T_{[>s]}) \cdot \theta_w \cdot \prod_{t \le l < s} (1-\rho_l) \cdot \rho_s \cdot I(T_{[t:s]} = w)\Big],$$

$$r_w(T) = \frac{1}{p(T)} \sum_{t=1}^{\tau_L} \Big[r_w(T_{[>t]}) \cdot I(T_{[1:t]} \ne w) + I(T_{[1:t]} = w)\Big] \cdot \theta_{T_{[\le t]}} \cdot \prod_{1 < l < t} (1-\rho_l) \cdot \rho_t \cdot p(T_{[>t]}),$$

$$\gamma_l(T) = \frac{p(T_{[\le l]}) \cdot p(T_{[>l]})}{p(T)},$$

where

$$p(T_{[t:s]}) = \mathbb{P}_\kappa(T_{[t:s]} \mid \mathcal{D}, \boldsymbol{\theta}) = \sum_{B \in \mathcal{B}_{[t:s]}} \mathbb{P}(T_{[t:s]} \mid B, \mathcal{D}, \boldsymbol{\theta}) \cdot \pi_\kappa(B),$$

with $\mathcal{B}_{[t:s]}$ being the truncated version of $\mathcal{B}$ according to the position window $[t:s]$.

As $p(T_{[<t]})$ and $p(T_{[>t]})$ can be derived in linear time via dynamic programming based on the following recursion:

$$p(T_{[<t]}) = \sum_{1 \le s \le \min(t-1, \tau_L)} \Big[p(T_{[<t-s]}) \cdot \theta_{T_{[t-s:t-1]}} \cdot \prod_{t-s \le l < t-1} (1-\rho_l) \cdot \rho_{t-1}\Big],$$

$$p(T_{[>t]}) = \sum_{1 \le s \le \min(L-t, \tau_L)} \Big[p(T_{[>t+s]}) \cdot \theta_{T_{[t+1:t+s]}} \cdot \prod_{t+1 \le l < t+s} (1-\rho_l) \cdot \rho_{t+s}\Big],$$

all computation issues involved can be efficiently resolved.

C Criteria for Masking PKU Corpus

For a specific text sequence $T = a_1 \cdots a_L \in \mathcal{T}_W$, let $B^* = (b_1^*, \cdots, b_L^*)$ be the standard segmentation adopted by the PKU corpus, while $B = (b_1, \cdots, b_L)$ be its word boundary profile predicted by a segmentation method $\mathcal{M}$. For each sub-string $S = a_{i_1} \cdots a_{i_2}$ of T, we say method $\mathcal{M}$ segments S with a coarser granularity with respect to B^* (denoted as $S \in \mathcal{G}_{\mathcal{M},B^*}$), if

$$b_{i_1-1} = b^*_{i_1-1} = 1 = b^*_{i_2} = b_{i_2}, \text{ and}$$

$$\sum_{i_1 < l < i_2} b_l = 0 \text{ and } \sum_{i_1 < l < i_2} b_l^* > 0.$$

Masking all sub-string $S \in \mathcal{G}_{\mathcal{M},B^*}$, we obtain the masked version of $\mathcal{T}_W$.