A Probabilistic Toolkit for Multi-grained Word Segmentation in Chinese

Xi Ma, Yang Hou, Xuebin Wang, Zhenghua Li*

School of Computer Science and Technology, Soochow University, China

{20225227115,yhou1,xbwan15}@stu.suda.edu.cn, zhli13@suda.edu.cn

Abstract

It is practically useful to provide consistent and reliable word segmentation results from different criteria at the same time, which is formulated as the multi-grained word segmentation (MWS) task. This paper describes a probabilistic toolkit for MWS in Chinese. We propose a new MWS approach based on the standard MTL framework. We adopt semi-Markov CRF for single-grained word segmentation (SWS), which can produce marginal probabilities of words during inference. For sentences that contain conflicts among SWS results, we employ the CKY decoding algorithm to resolve conflicts. Our resulting MWS tree can provide the criteria information of words, along with the probabilities. Moreover, we follow the works in SWS, and propose a simple strategy to exploit naturally annotated data for MWS, leading to substantial improvement of MWS performance in the cross-domain scenario.

1 Introduction

Given an input sentence consisting of n characters, denoted as $\boldsymbol{x} = c_0c_1...c_{n-1}$, the goal of word segmentation (WS) is to produce a word sequence, denoted as $\boldsymbol{y} = w_0w_1 \ldots w_{m-1}$, where $w_k = c_i \ldots c_j$ represents a word, which is also denoted as (i, j) afterwards.

Since words are the basic units for expressing conception or meaning, WS is fundamental for tasks like syntactic parsing, semantic parsing, information extraction, etc. Over the past decade, thanks to the development of deep learning, especially of pre-trained language models like BERT (Devlin et al., 2019), research on WS has made great progress (Wang et al., 2018; Zhao et al., 2018b; Shi et al., 2019; Yang, 2019; Li et al., 2023; Xu, 2024),

Meanwhile, there exist multiple WS criteria that follow different linguistic theories or target different scenarios in which WS results are required. For each criterion, WS data are manually annotated with great effort. In practice, it is often challenging to choose an appropriate WS criterion when utilizing WS results. Current works provide two directions for addressing this issue.

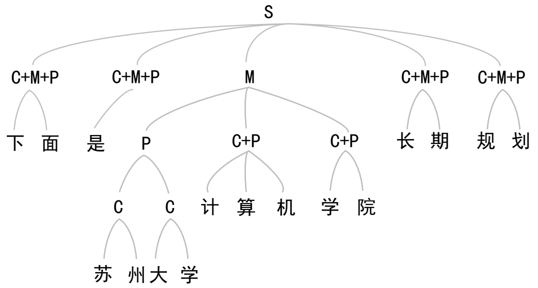


Figure 1: An MWS tree produced by our demo. Word-by-word translation is: “下面(below) 是(is) 苏州大学(Soochow University) 计算机(computer) 学院(department) 长期(long-term) 规划(plan)”. The labels of non-terminal nodes give which criteria each word comes from, in the descending order of marginal probabilities in the three SWS results (C for CTB, M for MSR, and P for PKU).

The first direction is the multi-criteria approach (Chen et al., 2017; Gong et al., 2019; Huang et al., 2020; Qiu et al., 2020; Chou et al., 2023). The basic idea is to leverage datasets from all criteria based on the multi-task learning (MTL) framework, in order to improve single-grained WS (SWS) performance of each individual criterion. Typically, the model contains a shared encoder, and separate decoders for each criterion.[1] During inference, the model can output all SWS results of all criteria given a sentence.

One crucial problem with the multi-criteria approach is that one SWS result ($\boldsymbol{y}^{\mathrm{a}}$) for one criterion

[*] Corresponding author. Email: zhli13@suda.edu.cn

[1] Qiu et al. (2020) share both a encoder and a decoder, but use an extra criterion embedding in the input layer to notify the model.

Proceedings of the 31st International Conference on Computational Linguistics: System Demonstrations, pages 83–90
January 19–24, 2025. ©2025 Association for Computational Linguistics

may conflict with that for another criterion ($\boldsymbol{y}^{\mathrm{b}}$). More specifically, $\boldsymbol{y}^{\mathrm{a}}$ contain a word that violates the boundaries of a word in $\boldsymbol{y}^{\mathrm{b}}$. For example, $(3, 6)$ conflicts with $(2, 5)$, and also with $(2, 4)$, but not with $(2, 8)$ nor $(3, 4)$.

As discussed in Gong et al. (2017), such conflicts are extremely rare in multi-criteria WS data, and when a word conflicts with another, it is almost certain that at least one of the two words is erroneous. This observation leads to the second direction, i.e., the multi-grained WS (MWS) task, which is formally proposed by Gong et al. (2017). MWS demands the model to resolve all conflicts, and produce a consistent hierarchical tree, in which non-terminal nodes correspond to word, as shown in Figure 1.

Gong et al. (2017), and the subsequent Gong et al. (2020), treat MWS as a constituent parsing problem. Due to the lack of annotated MWS data, they construct pseudo training data by performing paired annotation conversion, upon three popular SWS data, i.e., the Penn Chinese Treebank (CTB) (Xue et al., 2005), the Microsoft Research Chinese Word Segmentation (MSR) corpus (Huang et al., 2006), and the People's Daily Corpus (PKU) from Peking University (Yu and Zhu, 1998). They also manually construct two test datasets, i.e., the in-domain NEWS-test, and the cross-domain BAIKE-test. However, their works may have two shortcomings. First, automatic annotation conversion itself is very challenging, and the resulting pseudo training data may contain noises. Second, their approach totally discards the criteria information, that is, which criteria contribute to each word in the resulting MWS tree, which may be useful in some scenarios.

This work follows the direction of Gong et al. (2017). We select three representative WS criteria with different grain sizes: CTB, MSR, and PKU. The CTB criterion adopts the finest-grained approach, while the MSR criterion represents the coarsest-grained one, typically treating entity information as single words. The PKU criterion maintains a medium-grained approach between these two extremes. These three different-grained segmentation methods correspond to three subtasks in our MTL framework. Based on this MTL framework, we propose a new MWS approach. For SWS, we employ semi-Markov CRF (semi-CRF), which can generate word-level marginal probabilities during inference. For sentences that contain conflicts among SWS results, which account for less than 11% of all test sentences, we employ the CKY decoding algorithm (Kasami, 1966; Younger, 1967) to resolve conflicts. Our resulting MWS tree can provide the criteria information of words, along with the probabilities. Moreover, we follow the works in SWS, and propose a simple strategy to exploit naturally annotated data for MWS, leading to substantial improvement of MWS performance in the cross-domain scenario.

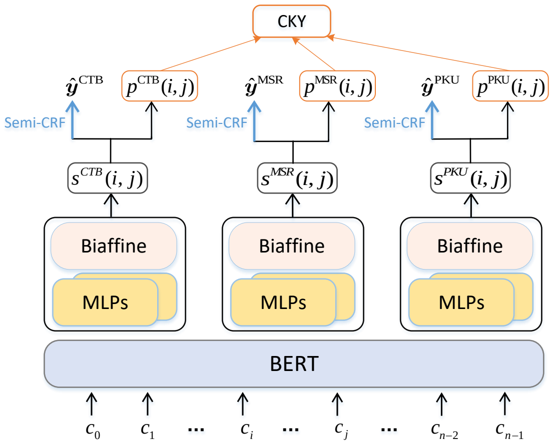


Figure 2: Model architecture.

We release our code package and pre-trained models at https://github.com/SUDA-LA/MWS-demo. Our proposed approach and code are independent in languages, and therefore can be applied to other languages lacking word delimiters such as Japanese and Korean.

2 MWS via MTL and CKY

Figure 2 gives the model architecture of our proposed approach. Under the MTL framework, three SWS submodels are trained, and can produce three SWS results in an independent manner during inference. Then, we employ the CKY algorithm to resolve the conflicts in the SWS results, producing a MWS tree.

2.1 Semi-CRF for SWS

In this work, we follow Liu et al. (2016) and employ semi-CRF (Sarawagi and Cohen, 2004) for SWS. Given scores of all spans, i.e., $s(\boldsymbol{x}, i, j)$ or shorten as $s(i, j)$, semi-CRF defines the score of a candidate segmentation $\boldsymbol{y}$ as:

$$s(\boldsymbol{x}, \boldsymbol{y}) = \sum_{(i,j)\in\boldsymbol{y}} s(i, j) \qquad (1)$$

In this sense, semi-CRF belongs to the family of span-based models, in contrast to the char-based sequence labeling models (Sutton et al., 2007; Papay et al., 2022)

As a probability model, semi-CRF then defines the conditional probability of $\boldsymbol{y}$ as:

$$p(\boldsymbol{y}|\boldsymbol{x}) = \frac{\exp(s(\boldsymbol{x},\boldsymbol{y}))}{Z(\boldsymbol{x}) \equiv \sum_{\boldsymbol{y}' \in \mathcal{Y}} \exp(s(\boldsymbol{x},\boldsymbol{y}'))} \quad (2)$$

where $Z(\boldsymbol{x})$ is the normalization term, and $\mathcal{Y}$ represents the set of all legal WS results.

Training loss. Given a mini-batch, i.e., $\mathcal{B} = \{\boldsymbol{x}_i, \boldsymbol{y}_i\}_{i=1}^{b}$, the loss is defined as:

$$\mathcal{L}(\mathcal{B}) = -\frac{1}{\#\texttt{word}} \times \sum_{i=1}^{b} \log p(\boldsymbol{y}_i|\boldsymbol{x}_i) \quad (3)$$

where $\#\texttt{word}$ is the total number of words in $\mathcal{B}$.

Inference. Semi-CRF aims to find the optimal segmentation using an efficient dynamic programming algorithm.

$$\hat{\boldsymbol{y}} = \operatorname*{argmax}_{\boldsymbol{y} \in \mathcal{Y}} s(\boldsymbol{x},\boldsymbol{y}) \quad (4)$$

The computational complexity is $O(n^2)$, but can be reduced to $O(Mn) = O(n)$ by constraining the maximum word length to a small constant, e.g., $M = 15$.

Marginal probabilities. One important feature of semi-CRF is that it can produce the marginal probability of candidate words.

$$p((i,j)|\boldsymbol{x}) = \sum_{(i,j) \in \boldsymbol{y} \in \mathcal{Y}} p(\boldsymbol{y}|\boldsymbol{x}) \quad (5)$$

Afterwards, we use $p(i,j)$ as a short form of $p((i,j)|\boldsymbol{x})$. Marginal probabilities are crucial for this work, as shown soon.

2.2 MTL-based Model Architecture

This work employs the MTL framework for MWS by treating each segmentation granularity as an individual task, as shown in Figure 2.

BERT as the shared encoder. They three tasks share the encoder. The parameters of BERT are fine-tuned during training, instead of frozen. For each character c_i in the input sentence, we use the output vector of the top layer of BERT as the contextual representation vector, i.e., $\boldsymbol{h}_i$.

Boundary representation and Biaffine scoring. We follow the constituency parsing work of Zhang et al. (2020), and employ MLPs to obtain boundary representation and a Biaffine component to compute scores of candidate spans. Each of the three tasks has separate MLPs and Biaffine component.

Inference. The three tasks independently produce optimal WS results, i.e., $\hat{\boldsymbol{y}}^{\mathsf{CTB}}$, $\hat{\boldsymbol{y}}^{\mathsf{MSR}}$, and $\hat{\boldsymbol{y}}^{\mathsf{PKU}}$. If the results have no conflicts, then we can build a hierarchical tree as shown in Figure 1, and consider it as the final MWS result.

Training. Each mini-batch is composed of sentences from the three training datasets, and three training losses are summed.

$$\mathcal{L}(.) = \mathcal{L}(\mathcal{B}^{\mathsf{CTB}}) + \mathcal{L}(\mathcal{B}^{\mathsf{MSR}}) + \mathcal{L}(\mathcal{B}^{\mathsf{PKU}}) \quad (6)$$

2.3 Resolving Conflicts via CKY over Marginal Probabilities

However, there may exist conflicts in the SWS results. For instance, $\hat{\boldsymbol{y}}^{\mathsf{CTB}}$ say that $(4,6)$ is a word, whereas $\hat{\boldsymbol{y}}^{\mathsf{MSR}}$ say $(3,7)$ is a word. Such overlapping makes it impossible to build a hierarchical tree, and is prohibited in MWS, as discussed in Section 1. In such circumstance, at least one of the two words must be erroneous and should be discarded.

Using our basic model, we find that the percentage of sentences having conflicts among SWS results is 1.7% in the in-domain NEWS-test data, and 10.9% in the cross-domain BAIKE-test data.

To resolve conflicts, we employ the CKY algorithm to produce a MWS tree. Please kindly note that we cannot directly use the scores of spans, i.e., $s(i,j)$, for CKY decoding. The reason is that the MLPs and Biaffines are independent for the three SWS tasks, and thus the scores are incomparable and may differ in the order of magnitude. Instead, we use the marginal probabilities, i.e., $p(i,j)$, as normalized scores. If a word appears in two SWS results, we choose the higher probability. For instance, if both $\hat{\boldsymbol{y}}^{\mathsf{CTB}}$ and $\hat{\boldsymbol{y}}^{\mathsf{MSR}}$ say that $(4,6)$ is word, with probabilities of 0.9 and 0.8 respectively. Then the normalized score of the word is 0.9 during CKY decoding.

Prior to decoding, we constrain the search space by modifying marginal probabilities [2]. For spans

[2] We conduct experiments comparing the performance with and without constraints on marginal probabilities. Results show that applying these constraints yields a 0.1 F-score improvement on the NEWS-test.

conflicting with existing SWS words, we set their probabilities to $-\infty$. For internal SWS conflicts, like $(1,3)$ and $(2,4)$, we treat their union $(1,4)$ as valid, setting probabilities of spans conflicting with $(1,4)$ to $-\infty$. This allows CKY decoding to resolve conflicts between $(1,3)$ and $(2,4)$, determining the correct segmentation.

The goal of CKY decoding is:

$$\hat{\boldsymbol{t}} = \underset{\boldsymbol{t}\in\mathcal{T}}{\operatorname{argmax}} \left(s(\boldsymbol{x},\boldsymbol{t}) \equiv \sum_{(i,j)\in\boldsymbol{t}} p(i,j) \right) \tag{7}$$

where $\boldsymbol{t}$ is a binarized tree. After obtaining $\hat{\boldsymbol{t}}$, we only detain words in $\hat{\boldsymbol{y}}^{\mathsf{CTB}} \cup \hat{\boldsymbol{y}}^{\mathsf{MSR}} \cup \hat{\boldsymbol{y}}^{\mathsf{PKU}}$ as the final MWS result.

For example, consider the anchor text fragment "生活水平线" (living standard line). Under the CTB criterion, it should be segmented as "生活|水平线" (life | standard line), while PKU criteria suggest "生活水平|线" (living standard | line). These conflicting word boundaries within the anchor text make it impossible to construct a proper hierarchical structure directly. Our proposed CKY decoding algorithm assigns a score to each candidate word within the anchor text fragment. In this case, the segmentation "生活|水平线" receives a higher probability score, leading to the final hierarchical structure "[[生活][水平线]]" ([[life][standard line]]).

3 Utilizing Naturally Annotated Data

Previous works successfully improve performance of SWS using naturally annotated data (Jiang et al., 2013; Liu et al., 2014; Zhao et al., 2018a). The basic assumption is that anchor texts in web pages are strong clues for word boundaries. Below is an example sentence containing an anchor text, omitting the invisible hyperlink.

下$_0$面$_1$是$_2$苏$_3$州大学计算机学院$_{11}$长$_{12}$期规划

We can see that $(3,11)$ correspond to an anchor text. Then, there should a word boundary between c_2 and c_3, and another word boundary between c_{11} and c_{12}. Any words that span any of the two boundary would produce conflicts, e.g., $(2,5)$, $(2,6)$, etc. In contrast, words like $(3,6)$ and $(7,9)$ do not conflict.

In this work, we utilize such naturally annotated data to further improve the performance of MWS. We collect about 12 million sentences with anchor texts from the Baidu Baike website[3] (abbreviated as BAIKE, similar to Wikipedia) after data cleaning. The major reason for using the BAIKE data instead of Wikipedia is that the evaluation data constructed by Gong et al. (2020) is also from BAIKE. We can directly see the effect of using naturally annotated data. Meanwhile, considering the broad genre coverage and large scale of BAIKE data, we expect that the improved model can obtain performance boost on a variety of texts, especially up-to-date texts.

Obtain partial MWS annotations. We apply the basic MWS model to BAIKE sentences without performing CKY decoding. Thus each sentence has three SWS results, corresponding to the three SWS criteria. To improve data quality, we discard sentences containing conflicts. We distinguish two types of conflicts. The first is that one SWS result conflicts with the boundaries of the anchor texts, and the second is that two SWS results contain conflicts.

After filtering sentences with conflicts, each sentence has three self-consistent SWS results. Then we only detain words inside the anchor text, and leave other parts of the sentence unsegmented. This is known as *partial annotation*. Taking the CTB criterion as an example, the resulting training sentence is:

下面是 /苏州/大学/计算机/学院/ 长期规划

Similarly, the PKU and MSR criteria respectively get one partially annotated sentence for training.

Training with partial annotation. Similar to linear-chain CRF (Liu et al., 2014), semi-CRF can be extended to accommodate such partially annotated sentences. One BAIKE sentence would receive three losses, corresponding to three SWS criteria, and we use their average as the final loss for the sentence.

4 Experiments

Data. The data used in this study is consistent with that used by Gong et al. (2020). It primarily comprises training sets from three annotation standards: CTB, MSR, and PKU, along with NEWS-dev, NEWS-test, and BAIKE-test datasets that have manually annotated multi-grained labels. Additionally, they employed a pseudo multi-grained labeled training data, referred to as Pseudo. Table 1 presents the statistics of these datasets.

[3] https://baike.baidu.com/

Settings. Following Gong et al. (2017), we use the standard evaluation metrics of F1 score, precision (P), and recall (R) to assess the performance of MWS.

We compared the performance of multiple methods using a fine-tuned BERT [4] (Devlin et al., 2019) as the encoder. The model configuration follows the setup described by Zhang et al. (2020). The training process for BERT involves 15 epochs, with early stopping applied based on performance on the development set.

4.1 Benchmark Methods

We employ four methods for comparison. Alongside the MTL method proposed in this work, we replicate two benchmark methods: tree parsing and single-task learning.

1. **Tree-based**: In the study by Gong et al. (2020), a span-based parser was trained using pseudo MWS data. Our replicate tree-based method aligns with the pseudo-labeled data they employ, and we extend their code by using BERT as the encoder.
2. **Separate**: Three SWS models are trained separately on the CTB, MSR, and PKU datasets. The results from three models are directly combined as the MWS results[5].
3. **Ours without CKY**: We employed a MTL framework to train three SWS submodels, enabling us to acquire MWS results according to three different criteria while preserving the criteria information of words.
4. **Ours with CKY**: Similar to **Ours without CKY**, but we introduced the CKY algorithm to resolve conflicts in the SWS results, thus generating the final MWS tree.

4.2 Main Result

Table 2 compares various methods on the NEWS-test and BAIKE-test datasets.

Comparison with baselines. We first compare our method with the single-task learning method (**Separate**) and the span-based parsing method (**Tree-based**) on NEWS-test and BAIKE-test datasets. Our observations indicate that **Separate** achieves relatively high recall compared to other methods, however, its precision is significantly lower due to its disregard for connections among different heterogeneous SWS data. The **Tree-based** model, conversely, attains relatively high precision at the expense of a lower recall. In contrast, the proposed method (**Ours without CKY**) demonstrates significant enhancements on both the NEWS-test and BAIKE-test datasets. It shows F1 score improvements of 0.2 and 3.12, 0.28 and 1.06 respectively, compared to these two baseline methods. These results underscore the suitability of our method for MWS tasks and its effectiveness in domain transfer.

Dataset		Annotation	#Sents	#Words	OOV(%)
Train	CTB	SWS	16,091	437,991	-
	MSR	SWS	78,226	2,121,758	-
	PKU	SWS	46,815	1,097,839	-
	Pseudo	MWS	138,628	4,127,461	-
Dev	NEWS	MWS	1,000	31,477	4.69
Test	NEWS	MWS	2,000	63,108	4.96
	BAIKE	MWS	6,320	14,450	40.71

Table 1: Data statistics in our experiments. Pseudo refers to automatically generated pseudo data. SWS and MWS stand for single-grained labels and multi-grained labels, respectively.

Model	NEWS-test			BAIKE-test		
	P	R	F1	P	R	F1
Gong20	95.24	90.59	92.86	48.39	38.91	43.14
Tree-based†	94.69	92.05	93.36	56.17	63.68	59.93
Separate	92.49	94.08	93.28	52.40	75.87	61.99
Ours (w/o CKY)	94.05	93.07	93.56	54.72	74.37	63.05
Ours	**95.26**	93.14	**94.19**	58.01	73.20	64.73
Adding BAIKE	94.40	93.73	94.06	**60.30**	**76.76**	**67.54**

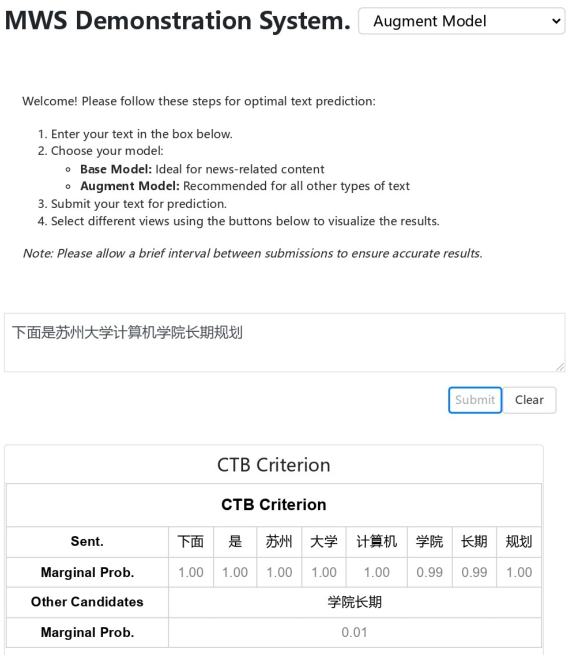
Table 2: The performance of different methods on the in-domain NEWS-test and the cross-domain BAIKE-test. Gong20 represents the work of Gong et al. (2020), which uses BiLSTM as the encoder. We modify their code to use BERT instead and retrain the model using the same training data, as indicated by †.

Impact of conflict resolution. We further investigate the impact of the conflict resolution strategy.[6] Compared to **Ours without CKY**, which simply overlooks conflicts, **Ours with CKY** shows notable performance enhancements. Our conflict resolution method demonstrates F1 score improvements of 0.63 and 1.68 on NEWS-test and BAIKE-test datasets, respectively. These results highlight the advantageous nature of conflict resolution in

[4] https://huggingface.co/bert-base-chinese

[5] Conflicting words are all included in the final results.

[6] According to our statistical analysis, 1.7% of the sentences in the NEWS test set and 10.9% of the sentences in the BAIKE test set contain conflicts.



MWS Demonstration System.

Welcome! Please follow these steps for optimal text prediction:

1. Enter your text in the box below.
2. Choose your model:
    - **Base Model:** Ideal for news-related content
    - **Augment Model:** Recommended for all other types of text
3. Submit your text for prediction.
4. Select different views using the buttons below to visualize the results.

Note: Please allow a brief interval between submissions to ensure accurate results.

下面是苏州大学计算机学院长期规划

CTB Criterion

CTB Criterion								
Sent.	下面	是	苏州	大学	计算机	学院	长期	规划
Marginal Prob.	1.00	1.00	1.00	1.00	1.00	0.99	0.99	1.00
Other Candidates	学院长期							
Marginal Prob.	0.01							

Figure 3: The SWS produced by our demo. Due to space limitations, only partial results are presented here; more detailed segmentation results are provided in the video demonstration we submit.

the MWS task. Ultimately, our method (**Ours with CKY**) outperforms the current SOTA model (**Tree-based**), achieving improvements of 0.83 and 4.8 on the two test datasets.

Analysis of Additional BAIKE Training Data Impact. To enhance the model's performance on cross-domain BAIKE-test, we introduced additional BAIKE training data to **Ours with CKY**. The data selection process followed the method described in Section 3, resulting in a refined dataset of 110,000 training samples. In the selected BAIKE sentences, the marginal probabilities of words in partially annotated sections ranged from 0.1 to 0.5. Experimental (**Adding BAIKE**) results demonstrate that incorporating this additional BAIKE training data significantly improved the model's cross-domain generalization capability. Specifically, We observe substantial improvements of 2.29, 3.56, and 2.81 in P, R, and F1 score, respectively. These findings underscore the crucial role of additional BAIKE data in enhancing the model's cross-domain adaptability.

5 System Overview

We encapsulate our trained model and provide both programmatic and graphical interfaces to support sentence prediction analysis.

Programmatic Interface. We encapsulate the trained model into a Python module named **Mws**. Researchers and developers can easily import and utilize this module with concise import statements. This modular approach enhances the model's portability and integrability, facilitating seamless integration into various Python projects and providing robust word segmentation support for downstream natural language processing tasks. Below is a partial output of sentence prediction using **Mws**. We provide a more detailed explanation of the usage of the **Mws** package in Appendix.

```
>>> from mws import Mws
>>> predictor=Mws()
>>> data=predictor.predict("下面是苏州大
学计算机学院长期规划")
>>> data.mws_res
[(0, 2),(2, 3),(3, 5),(3, 7),(3, 12),
(5, 7),(7, 10),(10, 12),(12, 14),(14, 16)]
>>> data.mws_prob
[1.0, 1.0, 0.49, 0.18, 0.33, 0.49,
0.6, 0.6, 0.99, 0.99]
```

Graphical User Interface. We develop a comprehensive web-based system to present the hierarchical structure of MWS. The backend is built with Flask, implementing a RESTful API for efficient communication. The interactive front-end, constructed using HTML, CSS, and JavaScript, allows users to input sentences, select model configurations, and view real-time prediction results. We employ the Fetch API for asynchronous communication with the backend. ECharts is utilized to render interactive tree diagrams, providing an intuitive visualization of MWS output. This architecture ensures a seamless and informative user experience for exploring MWS results.

The tree diagram, as shown in Figure 1, displays the hierarchical structure of MWS results and word criteria information. Leaf nodes represent characters, while non-leaf nodes indicate the criteria source of each word. For example, "苏州大学计算机学院" (School of Computer Science and Technology, Soochow University) is annotated as MSR, with "苏州大学" (Soochow University) segmented by PKU, "计算机" (computer) and "学院" (department) segmented by CTB and PKU, and "苏州" (Suzhou) and "大学" (University) by CTB.

The interface, illustrated in Figure 3, allows users to view SWS results based on different annotation criteria (MSR, CTB, PKU) and displays candidate words with marginal probabilities exceeding 0.01. This comprehensive view facilitates in-depth analysis of model behavior.

Performance Analysis. We evaluate the system's performance from both programmatic and web-based interfaces to assess its real-time application capabilities. For the programmatic interface, our model achieves a prediction speed of 40 sentences per second on a GPU (1080Ti) server, which meets the requirements of most real-time applications, such as text preprocessing in NLP pipelines and online document analysis. Our lightweight model design enables easy deployment on standard servers or integration into larger systems. For the web interface, we have implemented request rate limiting and response caching mechanisms to ensure system stability and optimal performance, maintaining responsive performance for real-time user interactions.

6 Conclusion

This work advances the state-of-the-art (SOTA) in MWS research through three key contributions. First, we apply span-based CWS methods to the MWS task, assessing our model on in-domain NEWS test data and cross-domain BAIKE test data. The MWS tree provides criteria information for words, and SWS offers more possible candidate words. Second, we introduce the CKY decoding algorithm to resolve segmentation conflicts, which significantly improved model performance. Our experiments demonstrate that this conflict resolution approach led to improvements of 0.63 and 1.68 F-scores on the NEWS-test and BAIKE-test, respectively. Finally, we explore the impact of data quality on model performance based on marginal probabilities and enhance the model's performance on cross-domain data by using a local loss function.

Acknowledgements

We thank the anonymous reviewers for their helpful comments and insights regarding our work. This work was supported by the National Natural Science Foundation of China (Grant NO. 62176173 and 62336006), and a project funded by the Priority Academic Program Development (PAPD) of Jiangsu Higher Education Institutions.

References

Xinchi Chen, Zhan Shi, Xipeng Qiu, and Xuanjing Huang. 2017. Adversarial multi-criteria learning for Chinese word segmentation. In *Proceedings of ACL*, pages 1193–1203.

Tzu Hsuan Chou, Chun-Yi Lin, and Hung-Yu Kao. 2023. Advancing Multi-Criteria Chinese Word Segmentation Through Criterion Classification and Denoising. In *Proceedings of ACL*, pages 6460–6476.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL*, pages 4171–4186.

Chen Gong, Zhenghua Li, Min Zhang, and Xinzhou Jiang. 2017. Multi-grained Chinese word segmentation. In *Proceedings of EMNLP*, pages 692–703.

Chen Gong, Zhenghua Li, Bowei Zou, and Min Zhang. 2020. Multi-grained Chinese word segmentation with weakly labeled data. In *Proceedings of COLING*, pages 2026–2036.

Jingjing Gong, Xinchi Chen, Tao Gui, and Xipeng Qiu. 2019. Switch-lstms for multi-criteria chinese word segmentation. *Proceedings of AAAI*, 33(01):6457–6464.

Chang-Ning Huang, Yumei Li, and Xiaodan Zhu. 2006. Tokenization guidelines of chinese text (v5. 0, in chinese). *Microsoft Research Asia*.

Weipeng Huang, Xingyi Cheng, Kunlong Chen, Taifeng Wang, and Wei Chu. 2020. Towards fast and accurate neural chinese word segmentation with multi-criteria learning. In *Proceedings of COLING*, pages 2062–2072.

Wenbin Jiang, Meng Sun, Yajuan Lü, Yating Yang, and Qun Liu. 2013. Discriminative Learning with Natural Annotations: Word Segmentation as a Case Study. In *Proceedings of ACL*, pages 761–769.

Tadao Kasami. 1966. An efficient recognition and syntax-analysis algorithm for context-free languages. *Coordinated Science Laboratory Report no. R-257*.

Hsiu-Wen Li, Ying-Jia Lin, Yi-Ting Li, Chun Lin, and Hung-Yu Kao. 2023. Improved unsupervised Chinese word segmentation using pre-trained knowledge and pseudo-labeling transfer. In *Proceedings of EMNLP*, pages 9109–9118.

Yijia Liu, Wanxiang Che, Jiang Guo, Bing Qin, and Ting Liu. 2016. Exploring segment representations for neural segmentation models. In *Proceedings of IJCAI*, page 2880–2886.

Yijia Liu, Yue Zhang, Wanxiang Che, Ting Liu, and Fan Wu. 2014. Domain adaptation for CRF-based Chinese word segmentation using free annotations. In *Proceedings of EMNLP*, pages 864–874.

Sean Papay, Roman Klinger, and Sebastian Padó. 2022. Constraining linear-chain crfs to regular languages. In *The Tenth International Conference on Learning Representations, ICLR 2022*.

Xipeng Qiu, Hengzhi Pei, Hang Yan, and Xuanjing Huang. 2020. A concise model for multi-criteria Chinese word segmentation with transformer encoder. In *Findings of EMNLP*, pages 2887–2897.

Sunita Sarawagi and William W. Cohen. 2004. Semi-markov conditional random fields for information extraction. In *Processing of NIPS 2004*, pages 1185–1192.

Xuewen Shi, Heyan Huang, Ping Jian, Yuhang Guo, Xiaochi Wei, and Yi-Kun Tang. 2019. Neural chinese word segmentation as sequence to sequence translation. *CoRR*, abs/1911.12982.

Charles Sutton, Andrew McCallum, and Khashayar Rohanimanesh. 2007. Dynamic conditional random fields: Factorized probabilistic models for labeling and segmenting sequence data. *J. Mach. Learn. Res.*, 8:693–723.

Dingmin Wang, Yan Song, Jing Li, Jialong Han, and Haisong Zhang. 2018. A hybrid approach to automatic corpus generation for Chinese spelling check. In *Proceedings of EMNLP*, pages 2517–2527.

Shiting Xu. 2024. BED: Chinese word segmentation model based on boundary-enhanced decoder. In *Proceedings of CACML*, pages 263–270.

Nianwen Xue, Fei Xia, Fu-Dong Chiou, and Martha Palmer. 2005. The Penn Chinese TreeBank: Phrase structure annotation of a large corpus. *Natural Language Engineering*, pages 207–238.

Haiqin Yang. 2019. BERT meets chinese word segmentation. *CoRR*, abs/1909.09292.

Daniel H. Younger. 1967. Recognition and Parsing of Context-Free Languages in Time n^3. *Information and Control*, pages 189–208.

Shiwen Yu and Xuefeng Zhu. 1998. Dictionary of Modern Chinese Grammar Information.

Yu Zhang, Houquan Zhou, and Zhenghua Li. 2020. Fast and accurate neural CRF constituency parsing. In *Proceedings of IJCAI*, pages 4046–4053.

Lujun Zhao, Qi Zhang, Peng Wang, and Xiaoyu Liu. 2018a. Neural networks incorporating unlabeled and partially-labeled data for cross-domain Chinese word segmentation. In *Proceedings of IJCAI*, pages 4602–4608.

Yue Zhao, Hang Li, Shoulin Yin, and Yang Sun. 2018b. A new Chinese word segmentation method based on maximum matching. *JIHMSP*, pages 1528–1535.

Appendix A: More Details on Module APIs

Upon inputting a Chinese sentence and invoking the programming interface, the system returns a comprehensive set of results. To access this segmentation service, users can download the project from our provided GitHub repository [1] and configure the local environment. Once being set up, the system outputs the following:

1. MWS results, accompanied by the probability of each word as determined by CKY decoding.
2. SWS results under three different annotation standards, along with their corresponding marginal probabilities.
3. For each annotation standard, additional candidate words with marginal probabilities exceeding 0.01, as derived from Semi-CRF decoding.

```
>>> from mws import Mws
>>> predictor=Mws()
>>> data=predictor.predict("下面是苏州大
学计算机学院长期规划")
>>> data.sentence
'下面是苏州大学计算机学院长期规划'
>>> data.mws_res
[(0, 2),(2, 3),(3, 5),(3, 7),(3, 12),
(5, 7),(7, 10),(10, 12),(12, 14),
(14, 16)]
>>> data.mws_prob
[1.0, 1.0, 0.49, 0.18, 0.33, 0.49,
0.6, 0.6, 0.99, 0.99]
>>> data.ctb_res
[(0, 2),(2, 3),(3, 5),(5, 7),(7, 10),
(10, 12),(12, 14),(14, 16)]
>>> data.ctb_prob
[1.0, 1.0, 1.0, 1.0, 1.0, 0.99, 0.99, 1.0]
>>> data.msr_res
[(0, 2),(2, 3),(3, 12),(12, 14),(14, 16)]
>>> data.msr_prob
[1.0, 1.0, 0.99, 0.99, 0.99]
>>> data.pku_res
[(0, 2),(2, 3),(3, 7),(7, 10),(10, 12),
(12, 14),(14, 16)]
>>> data.pku_prob
[1.0, 1.0, 0.54, 0.8, 0.8, 0.98, 0.98]
>>> data.ctb_cand
[(11,15)]
>>> data.msr_cand
[(3,12)]
>>> data.pku_cand
[]
```