

# Dual Long Short-Term Memory Networks for Sub-Character Representation Learning

Han He<sup>1</sup>, Lei Wu<sup>2</sup>, Xiaokun Yang<sup>3</sup>, Hua Yan<sup>4</sup>  
Zhimin Gao<sup>5</sup>, Yi Feng<sup>6</sup>, George Townsend<sup>7</sup>

<sup>1,2,3,4</sup>University of Houston-Clear Lake, U.S.A

<sup>5</sup>University of Houston, U.S.A <sup>6,7</sup>Alsomu University, Canada

<sup>1,2,3,4</sup>{heh1996, wul, yangxia, yan}@uhcl.edu

<sup>5</sup>zgao5@uh.edu, <sup>6,7</sup>{feng, townsend}@algomau.ca

*Abstract*—Characters have commonly been regarded as the minimal processing unit in Natural Language Processing (NLP). But many non-latin languages have hieroglyphic writing systems, involving a big alphabet with thousands or millions of characters. Each character is composed of even smaller parts, which are often ignored by the previous work. In this paper, we propose a novel architecture employing two stacked Long Short-Term Memory Networks (LSTMs) to learn sub-character level representation and capture deeper level of semantic meanings. To build a concrete study and substantiate the efficiency of our neural architecture, we take Chinese Word Segmentation as a research case example. Among those languages, Chinese is a typical case, for which every character contains several components called radicals. Our networks employ a shared radical level embedding to solve both Simplified and Traditional Chinese Word Segmentation, without extra Traditional to Simplified Chinese conversion, in such a highly end-to-end way the word segmentation can be significantly simplified compared to the previous work. Radical level embeddings can also capture deeper semantic meaning below character level and improve the system performance of learning. By tying radical and character embeddings together, the parameter count is reduced whereas semantic knowledge is shared and transferred between two levels, boosting the performance largely. On 3 out of 4 Bakeoff 2005 datasets, our method surpassed state-of-the-art results by up to 0.4%. Our results are reproducible, source codes and corpora are available on GitHub<sup>1</sup>.

*Index Terms*—AI Algorithms and Applications, Deep Learning, Machine Learning Algorithms, Natural Language Processing, Neural Networks, Pattern Recognition

TABLE I  
ILLUSTRATION OF SEMANTIC COMPONENT (SEM.) AND PHONETIC COMPONENT (PHO.) IN SIMPLIFIED CHINESE (SC) AND TRADITIONAL CHINESE (TC).

SC	Sem.	Pho.	TC	Sem.	Pho.
鲤	鱼	里	鯉	魚	里
鲢	鱼	连	鯪	魚	連
河	水	可	河	水	可
沟	水	勾	溝	水	冓
捞	手	劳	撈	手	勞
捡	手	金	撿	手	僉

## I. INTRODUCTION

Unlike English, the alphabet in many non-latin languages is often big and complex. In those hieroglyphic writing systems, every character can be decomposed into smaller parts or sub-characters, and each part has special meanings. But existing methods often follow common processing steps in latin flavor [1]–[5], and treat character as the minimal processing unit, leading to a neglecting of information inside non-latin characters. Early work exploiting sub-character information usually treat it as a separate level from character [6]–[9], ignoring the language phenomenon that some of those sub-characters themselves are often used as normal characters. From this phenomenon, we gained a new motivation to design a novel neural network architecture for learning character and sub-character representation jointly.

In linguists’ view, Chinese writing system is such a highly hieroglyphic language, and it has a long history of character compositionality. Every Chinese character has several radicals (“部首” in Chinese),

<sup>1</sup><https://github.com/hankcs/sub-character-cws>

which serves as semantic component for encoding meaning, or phonetic component for representing pronunciation. For instance, we listed radicals of several Simplified and Traditional Chinese characters in Table I. Chinese characters with same semantic component are closely correlated in semantic. As shown above, carp (鲤) and silverfish (鲢) are both fish (鱼). River (河) and gully (沟) are all filled with water (水). To catch (捞) or to pick up (捡) a fish, one needs to use hands (手). To exploit those semantic meanings under character embedding level, radical embedding emerged since 2014 [6], [8]–[10]. These early work treated sub-character and character as two separate levels, omitting that they can actually be unified as single minimal processing unit in language model. Instead of ignoring linguistic knowledge, we respect the divergence of human language, and propose a novel joint learning framework for both character and sub-character representations.

To verify the efficiency of our jointly learnt representations, we conducted extensive experiments on the Chinese Word Segmentation (CWS) task. As those languages often don't have explicit delimiters between words, making it hard to perform later NLP tasks like Information Retrieval or Question Answering. Chinese language is such a typical non-segmented language, which means unlike English language having spaces between every word, Chinese has no explicit word delimiters. Therefore, Chinese Word Segmentation is a preliminary pre-processing step for later Chinese language process tasks. Recently with the rapid rise of deep learning, neural word segmentation approaches arose to reduce efforts in feature engineering [11]–[16].

In this paper, we propose a novel model to dive deeper into character embeddings. In our framework, Simplified Chinese and Traditional Chinese corpora are unified via radical embedding, growing an end-to-end model. Every character is converted to a sequence of radicals with its original form. Character embeddings and radical embeddings are pretrained jointly in Bojanowski et al. [3]'s subword aware method. Finally, we conducted various experiments on corpora from SIGHAN bakeoff 2005. Results showed that our jointly learnt character embedding outperforms conventional character embedding training methods. Our models can improve

performance by transfer learning between characters and radicals. The final scores surpassed previous work, and 3 out of 4 even surpassed previous preprocessing-heavy state-of-the-art learning work.

More specifically, the contributions of this paper could be summarized as:

- Explored a novel sub-character aware neural architecture and unified character and sub-character as one same level embedding.
- Released the first full Chinese character-radical conversion corpus along with pre-trained embeddings, which can be easily applied on other NLP tasks. Our codes and corpora are freely available for the public.

## II. RELATED WORK

In this section, we review the previous work from 2 directions – radical embedding and Chinese Word Segmentation.

### A. Radical Embedding

To leverage the semantic meaning inside Chinese characters, Sun et al. [6] inaugurated radical information to enrich character embedding via softmax classification layer. In similar way, Li et al. [7] proposed charCBOW model taking concatenation of the character-level and component-level context embeddings as input. Making networks deeper, Shi et al. [8] proposed a deep CNN on top of radical embedding pre-trained via CBOW. Instead of utilizing CNNs, following Lample et al. [17], Dong et al. [9] used two level LSTMs taking character embedding and radical embedding as input respectively.

Our work is closely related to Dong et al. [9], but there are two major differences. In pre-training phase, their character embeddings were pre-trained separately, by utilizing conventional word2vec package, and the radical embeddings are randomly initialized. While we considered radical units as sub-characters (parts of one character) and trained the two level embeddings jointly, following Bojanowski et al. [3]'s approach. In training and testing phases, our two-level embeddings are tied up and unified as the sole minimal input unit of Chinese language.

### B. Chinese Word Segmentation

Chinese Word Segmentation has been a well-known NLP task for decades [18]. After pioneer

Xue et al. [19] transformed CWS into a character-based tagging problem, Peng et al. [20] adopted CRF as the sequence labeling model and showed its effectiveness. Following these pioneers, later sequence labeling based work [21]–[24] was proposed. Recent neural models [9], [11], [13], [14], [25], [26] also followed this sequence labeling fashion.

Our model is based on Bi-LSTM with CRF as top layer. Unlike previous approaches, the inputs to our model are both character and radical embeddings. Furthermore, we explored which embedding level is more tailored for Chinese language, either using both embeddings together, or even tying them up.

### III. JOINT LEARNING FOR CHARACTER EMBEDDING AND RADICAL EMBEDDING

Previous work treated character and radical as two different levels, used them separately or used one to enhance the other. Although radicals are components of a character (belonging to a lower level), they can actually be learnt jointly. It is linguistically more reasonable to put radical embeddings and character embeddings in exactly the same vector space. We propose to train character vector representation being aware of its internal structure of radicals.

#### A. Character Decomposition

Every character can be decomposed into a list of radicals or components. To maintain character information in radical list, we simply add the raw form of character to its radical list. Taking the linguistic knowledge that semantic component contains richest meaning of one character into consideration, we append the semantic component to the end of its radical list, hence to make the semantic component appear more than once.

Formally, denote  $c$  as a character,  $r$  as a radical,  $\mathcal{L}_c = [r_1, r_2 \cdots r_n]$  as the original radical list of  $c$ . Let  $r_s \in \mathcal{L}_c$  be the semantic component of  $c$ . Our decomposition of  $c$  will be:

$$\mathcal{R}_c = [c, r_1, r_2 \cdots r_n, r_s] \quad (1)$$

#### B. General Continuous Skip-Gram (SG) Model

Take a brief review of the continuous skip-gram model introduced by Mikolov et al. [10], applied in character representation learning.

Given an alphabet, target is to learn a vectorial representation  $\mathbf{v}_c$  for each character  $c$ . Let  $c_1, \dots, c_T$  be a large-scale corpus represented as a sequence of characters, the objective function of the skipgram model is to maximize the log-likelihood of correct prediction. The probability of a context character  $c_y$  given  $c_x$  is computed by a scoring function  $s$  which maps character and context to scores in  $\mathbb{R}$ .

The general SG model ignores the radical structure of characters, we propose a different scoring function  $s$ , in order to capture radical information.

Let all radicals form an alphabet of size  $R$ . Given a character  $c$  and the radical list  $\mathcal{R}_c \subset \{1, \dots, R\}$  of  $c$ , a vector representation  $\mathbf{z}_r$  is associated to each radical  $r$ . Then a character is represented by the sum of the vector representations of its radicals. Thus the new scoring function will be:

$$s(c_x, c_y) = \sum_{r \in \mathcal{R}_{c_x}} \mathbf{z}_r^\top \mathbf{v}_{c_y}. \quad (2)$$

This simple model allows learning the representations of characters and radicals jointly.

### IV. RADICAL AWARE NEURAL ARCHITECTURES FOR GENERAL CHINESE WORD SEGMENTATION

Once character and radical representations are learnt, one evaluation metric is how much it improves a NLP task. We choose the Chinese Word Segmentation task as a standard benchmark to examine their efficiency. One prevailing approach to CWS is casting it to character based sequence tagging problem, where our representations can be applied. A commonly used tagging set is  $\mathcal{T} = \{B, M, E, S\}$ , representing the **begin**, **middle**, **end** of a word, or single character forming a word.

Given a sequence  $\mathbf{X}$  consisted of  $n$  features as  $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n)$ , the goal of sequence tagging based CWS is to find the most possible tags  $\mathbf{Y}^* = \{\mathbf{y}_1^*, \dots, \mathbf{y}_n^*\}$ :

$$\mathbf{Y}^* = \arg \max_{\mathbf{Y} \in \mathcal{T}^n} p(\mathbf{Y} | \mathbf{X}), \quad (3)$$

where  $\mathcal{T} = \{B, M, E, S\}$ .

Since tagging set restricts the order of adjacent tags, we model them jointly using a conditional random field, mostly following the architecture proposed by Lample et al. [17], via stacking two LSTMs with a CRF layer on top of them.

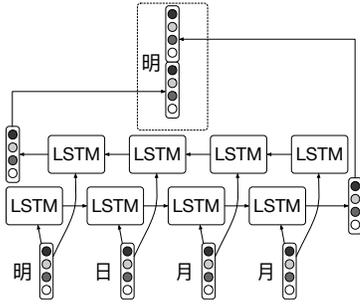


Fig. 1. Radical LSTM Layer – composition of character representation from radicals

### A. Radical LSTM Layer: Character Composition from Radicals

In this section, we’ll review RNN with Bi-LSTM extension briefly, before introducing our character composition network.

a) *LSTM*: Long Short-Term Memory Networks (LSTMs) [27] are extensions of Recurrent Neural Networks (RNNs). They are designed to combat gradient vanishing issue via incorporating a memory-cell which enables long-range dependencies capturing.

b) *Bi-LSTM*: One LSTM can only produce the representation  $\vec{h}_t$  of the left context at every character  $t$ . To incorporate a representation of the right context  $\overleftarrow{h}_t$ , a second LSTM which reads the same sequence in reverse order is used. Pair of this forward and backward LSTM is called bidirectional LSTM (Bi-LSTM) [28] in literature. By concatenating its left and right context representations, the final representation is produced as  $\mathbf{h}_t = [\vec{h}_t; \overleftarrow{h}_t]$ .

We apply a Bi-LSTM to compose character embeddings from radical embeddings in both directions. The raw character is inserted as the first radical, and the semantic component is appended as the last radical. The motivation behind this trick is to make use of LSTM’s bias phenomena. In practice, LSTMs usually tend to be biased towards the most recent inputs of the sequence, thus the first one or last one depends on its direction.

As illustrated in Figure 1, the character 明 (bright) has the radical list of 日 (sun) and 月 (moon) with its raw form and duplicated semantic radical. Its compositional representation  $\mathbf{h}_i^r \in \mathbb{R}^{2k}$  is agglomerated via a Bi-LSTM from these radical embeddings, where  $k$  is the dimension of radical embeddings.

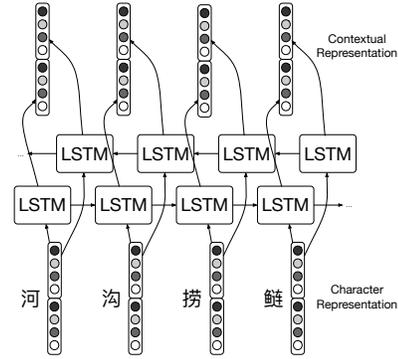


Fig. 2. Character LSTM Layer – capture contextual representation

### B. Character Bi-LSTM Layer: Context Capturing

Once compositional character representation  $\mathbf{h}_i^r$  is synthesized, the contextual representation  $\mathbf{h}_t^c \in \mathbb{R}^{2d}$  at every character  $t$  in input sentence can be agglomerated by a second Bi-LSTM. The dimension  $d$  is a flexible hyper-parameter, which will be explored in later experiments.

Our architecture for contextual feature capturing is shown in Figure 2. This contextual feature vector contains the meaning of a character, its radicals and its context.

### C. CRF Layer: Tagging Inference

We employed a Conditional Random Fields (CRF) [29] layer as the inference layer. As first order linear chain CRFs only model bigram interactions between output tags, so the maximum of a posteriori sequence  $\mathbf{Y}^*$  in Eq. 3 can be computed using dynamic programming, both in training and decoding phase. The training goal is to maximize the log-probability of the gold tag sequence.

## V. EXPERIMENTS

We conducted various experiments to verify the following questions:

- 1) Does radical embedding enhance character embedding in pre-training phase?
- 2) Whether radical embedding helps character embedding in training phase and test phase (by using character embedding solely or using them both)?
- 3) Can radical embedding replace character embedding (by using radical embedding only)?
- 4) Should we tie up two level embeddings?

## A. Datasets

To explore these questions, we experimented on the 4 prevalent CWS benchmark datasets from SIGHAN2005 [30]. Following conventions, the last 10% sentences of training set are used as development set.

## B. Radical Decomposition

We obtained radical lists of character from the *online Xinhua Dictionary*<sup>2</sup>, which are included in our open-source project.

## C. Pre-training

Previous work have shown that pre-trained embeddings on large unlabeled corpus can improve performance. It usually involves lots of efforts to preprocess those corpus. Here we presented a novel solution.

The corpus used is Chinese Wikipedia of July 2017. Unlike most approaches, we don't perform Traditional Chinese to Simplified Chinese conversion. Our radical decomposition is sufficient of associate character to its similar variants. Not only traditional-simplified character pairs, those with similar radical decompositions will also share similar vectorial representations.

Further, instead of the commonly used word2vec [2], we utilized fastText<sup>3</sup> [3] to train character embeddings and radical embeddings jointly. We applied SG model, 100 dimension, and set both maximum and minimal  $n$ -gram length to 1, as the radical takes only one token.

## D. Final Results on SIGHAN bakeoff 2005

Our baseline model is Bi-LSTM-CRF trained on each datasets only with pre-trained character embedding (the conventional word2vec), no sub-character enhancement, no radical embeddings. Then we improved it with sub-character information, adding radical embeddings, tying two level embeddings up. The final results are shown in Table II.

All experiments are conducted with standard Bakeoff scoring program<sup>4</sup> calculating precision, re-

<sup>2</sup><http://tool.httpcn.com/Zi/>

<sup>3</sup><https://github.com/facebookresearch/fastText> With tiny modification to output  $n$ -gram vectors.

<sup>4</sup><http://www.sighan.org/bakeoff2003/score> This script rounds a score to one digit.

TABLE II

COMPARISON WITH PREVIOUS STATE-OF-THE-ART MODELS OF RESULTS ON ALL FOUR BAKEOFF-2005 DATASETS.

Models	PKU	MSR	CityU	AS
Tseng et al. [21]	95.0	96.4	-	-
Zhang and Clark [31]	95.0	96.4	-	-
Sun et al. [32]	95.2	97.3	-	-
Sun et al. [24]	95.4	<b>97.4</b>	-	-
Pei et al. [13]	95.2	97.2	-	-
Chen et al. [26]	94.3	96.0	95.6	94.8
Cai et al. [16] <sup>◇</sup>	<b>95.8</b>	97.1	95.6	95.3
baseline	94.6	96.0	94.7	94.8
+subchar	95.0	96.0	94.9	94.9
+radical	94.6	96.7	95.3	95.2
+radical -char	94.4	96.5	95.0	95.1
+radical +tie	94.8	96.8	95.3	95.1
+radical +tie +bigram	95.3	<b>97.4</b>	<b>95.9</b>	<b>95.7</b>

call, and  $F_1$ -score. Note that results with  $\diamond$  expurgated long words in test set.

## E. Model Analysis

Sub-character information enhances character embeddings. Previous work showed pre-trained character embeddings can improve performance. Our experiment showed with sub-character information (+subchar), performance can be further improved compared to no sub-character enhancement (baseline). By simply replacing the conventional word2vec embeddings to radical aware embeddings, the score can benefit an improvement as much as 0.4%.

Radical embeddings collaborate well with character embeddings. By building compositional embeddings from radical level (+radical), performance increased by up to 0.7% in comparison with model (baseline) on MSR dataset. But we also notice that: 1) On small dataset such as PKU, radical embeddings cause tiny performance drop. 2) With the additional bigram feature, performance can be further increased as much as 0.6%.

Radical embeddings can't fully replace character embeddings. Without character embeddings but use radical embeddings solely (+radical -char), performance drops a little (0.1% to 0.3%) compared to the model with character embeddings (+radical).

Tying two level embeddings up is a good idea. By tying radical embeddings and character embeddings together (+radical +tie), the raw feature is unified into the same vector space, knowledge is transferred between two levels, and performance is boosted up to 0.2%.

## VI. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed a novel neural network architecture with dedicated pre-training techniques to learn character and sub-character representations jointly. As an concrete application example, we unified Simplified and Traditional Chinese characters through sub-character or radical embeddings. We have utilized a practical way to train radical and character embeddings jointly. Our experiments showed that sub-character information can enhance character representations for a pictographic language like Chinese. By using both level embeddings and tying them up, our model has gained the most benefit and surpassed previous single criterial CWS systems on 3 datasets.

Our radical embeddings framework can be applied to extensive NLP tasks like POS-tagging and Named Entity Recognition (NER) for various hieroglyphic languages. These tasks will benefit from deeper level of semantic representations encoded with more linguistic knowledge.

## REFERENCES

- [1] T. Mikolov, M. Karafiát, L. Burget, J. Cernocký, and S. Khudanpur, "Recurrent neural network based language model." in *Interspeech*, vol. 2, 2010, p. 3.
- [2] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," *arXiv.org*, Jan. 2013.
- [3] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching Word Vectors with Subword Information," *arXiv.org*, p. arXiv:1607.04606, Jul. 2016.
- [4] Y. Kim, Y. Jernite, D. Sontag, and A. M. Rush, "Character-aware neural language models." in *AAAI*, 2016, pp. 2741–2749.
- [5] Y. Pinter, R. Guthrie, and J. Eisenstein, "Mimicking word embeddings using subword rnns," *arXiv preprint arXiv:1707.06961*, 2017.
- [6] Y. Sun, L. Lin, N. Yang, Z. Ji, and X. Wang, "Radical-Enhanced Chinese Character Embedding." *ICONIP*, vol. 8835, no. Chapter 34, pp. 279–286, 2014.
- [7] Y. Li, W. Li, F. Sun, and S. Li, "Component-Enhanced Chinese Character Embeddings." *EMNLP*, 2015.
- [8] X. Shi, J. Zhai, X. Yang, Z. Xie, and C. Liu, "Radical Embedding - Delving Deeper to Chinese Radicals." *ACL*, 2015.
- [9] C. Dong, J. Zhang, C. Zong, M. Hattori, and H. Di, "Character-Based LSTM-CRF with Radical-Level Features for Chinese Named Entity Recognition." *NLPCC/ICCPOL*, 2016.
- [10] T. Mikolov, I. Sutskever, K. C. 0010, G. S. Corrado, and J. Dean, "Distributed Representations of Words and Phrases and their Compositionality." *NIPS*, 2013.
- [11] X. Zheng, H. Chen, and T. Xu, "Deep Learning for Chinese Word Segmentation and POS Tagging." *EMNLP*, 2013.
- [12] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. P. Kuksa, "Natural Language Processing (Almost) from Scratch." *Journal of Machine Learning Research*, 2011.
- [13] W. Pei, T. Ge, and B. Chang, "Max-Margin Tensor Neural Network for Chinese Word Segmentation." *ACL*, 2014.
- [14] X. Chen, X. Qiu, C. Zhu, P. Liu, and X. Huang, "Long Short-Term Memory Neural Networks for Chinese Word Segmentation." *EMNLP*, 2015.
- [15] D. Cai and H. Zhao, "Neural Word Segmentation Learning for Chinese." *ACL*, 2016.
- [16] D. Cai, H. Zhao, Z. Zhang, Y. Xin, Y. Wu, and F. Huang, "Fast and Accurate Neural Word Segmentation for Chinese," *arXiv.org*, p. arXiv:1704.07047, Apr. 2017.
- [17] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, "Neural Architectures for Named Entity Recognition." *CoRR*, 2016.
- [18] C. Huang and H. Zhao, "Chinese word segmentation: A decade review," *Journal of Chinese Information Processing*, vol. 21, no. 3, pp. 8–19, 2007.
- [19] N. Xue, "Chinese Word Segmentation as Character Tagging." *IJCLCLP*, 2003.
- [20] F. Peng, F. Feng, and A. McCallum, "Chinese segmentation and new word detection using conditional random fields," pp. 562–568, 2004.
- [21] H. Tseng, P. Chang, G. Andrew, D. Jurafsky, and C. Manning, "A conditional random field word segmenter for sighthan bakeoff 2005," 2005, pp. 168–171.
- [22] H. Zhao, C. Huang, M. Li, and B.-L. Lu, "Effective Tag Set Selection in Chinese Word Segmentation via Conditional Random Field Modeling." *PACLIC*, 2006.
- [23] H. Zhao, C. N. Huang, M. Li, and B. L. Lu, "A unified character-based tagging framework for chinese word segmentation," *Acm Transactions on Asian Language Information Processing*, vol. 9, no. 2, pp. 1–32, 2010.
- [24] X. Sun, H. Wang, and W. Li, "Fast online training with frequency-adaptive learning rates for chinese word segmentation and new word detection," pp. 253–262, 2012.
- [25] Y. Qi, S. G. Das, R. Collobert, and J. Weston, "Deep Learning for Character-Based Information Extraction." *ECIR*, 2014.
- [26] X. Chen, Z. Shi, X. Qiu, and X. Huang, "Adversarial Multi-Criteria Learning for Chinese Word Segmentation." vol. 1704, p. arXiv:1704.07556, 2017.
- [27] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [28] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures," *Neural Networks*, vol. 18, no. 5-6, pp. 602–610, Jul. 2005.
- [29] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Eighteenth International Conference on Machine Learning*, 2001, pp. 282–289.
- [30] T. Emerson, "The second international chinese word segmentation bakeoff," in *Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing*. Jeju Island, Korea, 2005, pp. 123–133.
- [31] Y. Zhang and S. Clark, "Chinese segmentation with a word-based perceptron algorithm." Prague, Czech Republic: Association for Computational Linguistics, Jun. 2007, pp. 840–847. [Online]. Available: <http://www.aclweb.org/anthology/P/P07/P07-1106>
- [32] X. Sun, Y. Zhang, T. Matsuzaki, Y. Tsuruoka, and J. Tsujii, "A discriminative latent variable chinese segmenter with hybrid word/character information," pp. 56–64, 2009.