Two Knives Cut Better Than One: Chinese Word Segmentation with Dual Decomposition

Abstract

There are two dominant approaches to Chinese word segmentation: word-based and character-based models, each with respective strengths. Prior work has shown that gains in segmentation performance can be achieved from combining these two types of models; however, past efforts have not provided a practical technique to allow mainstream adoption. We propose a method that effectively combines the strength of both segmentation schemes using an efficient dual-decomposition algorithm for joint inference. Our method is simple and easy to implement. Experiments on SIGHAN 2003 and 2005 evaluation datasets show that our method achieves the best reported results to date on 6 out of 7 datasets.

1 Introduction

Chinese text is written without delimiters between words; as a result, Chinese word segmentation (CWS) is an essential foundational step for many tasks in Chinese natural language processing. As demonstrated by (Shi and Wang, 2007; Bai et al., 2008; Chang et al., 2008; Kummerfeld et al., 2013), the quality and consistency of segmentation has important downstream impacts on system performance in machine translation, POS tagging and parsing.

State-of-the-art performance in CWS is high, with F-scores in the upper 90s. Still, challenges remain. Unknown words, also known as out-of-vocabulary (OOV) words, lead to difficulties for word- or dictionary-based approaches. Ambiguity can cause errors when the appropriate segmentation is determined contextually, such as 才能 ("talent") and 才 / 能 ("just able") (Gao et al., 2003).

There are two primary classes of models: character-based, where the foundational units for processing are individual Chinese characters (Xue, 2003; Tseng et al., 2005; Zhang et al., 2006; Wang et al., 2010), and word-based, where the units are full words based on some dictionary or training lexicon (Andrew, 2006; Zhang and Clark, 2007). Sun (2010) details their respective theoretical strengths: character-based approaches better model the internal compositional structure of words and are therefore more effective at inducing new OOV words; word-based approaches are better at reproducing the words of the training lexicon and can capture information from significantly larger contextual spans. Prior work has shown performance gains from combining these two types of models to exploit their respective strengths, but such approaches are often complex to implement and computationally expensive.

In this work, we propose a simple and principled joint decoding method for combining character-based and word-based segmenters based on dual decomposition. This method has strong optimality guarantees and works very well empirically. It is easy to implement and does not require retraining of existing character- and word-based segmenters. Perhaps most importantly, this work presents a much more practical and usable form of classifier combination in the CWS context than existing methods offer.

Experimental results on standard SIGHAN 2003 and 2005 bake-off evaluations show that our model outperforms the character and word baselines by a significant margin. In particular, out approach improves OOV recall rates and segmentation consistency, and gives the best reported results to date on

6 out of 7 datasets.

2 Models for CWS

In this section, we describe the character-based and word-based models we use as baselines, review existing approaches to combination, and describe our algorithm for joint decoding with dual decomposition.

2.1 Character-based Models

In the most commonly used contemporary approach to character-based segmentation, first proposed by (Xue, 2003), CWS is seen as a character sequence tagging task, where each character is tagged on whether it is at the beginning, middle, or end of a word. Conditional random fields (CRF) (Lafferty et al., 2001) have been widely adopted for this task, and give state-of-the-art results (Tseng et al., 2005). In a first-order linear-chain CRF model, the conditional probability of a label sequence $\mathbf{y}$ given a word sequence $\mathbf{x}$ is defined as:

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z}\sum_{t=1}^{|\mathbf{y}|}\exp\left(\theta \cdot f(x, y_t, y_{t+1})\right)$$

$f(x, y_t, y_{t-1})$ are feature functions that typically include surrounding character n-gram and morphological suffix/prefix features. These types of features capture the compositional properties of characters and are likely to generalize well to unknown words. However, the Markov assumption in CRF limits the context of such features; it is difficult to capture long-range word features in this model.

2.2 Word-based Models

Word-based models search through lists of word candidates using scoring functions that directly assign scores to each. Early word-based segmentation work employed simple heuristics like dictionary-lookup maximum matching (Chen and Liu, 1992). More recently, Zhang and Clark (2007) reported success using a linear model trained with the average perceptron algorithm (Collins, 2002). Formally, given input $\mathbf{x}$, their model seeks a segmentation $F(\mathbf{x})$ such that:

$$F(\mathbf{y}|\mathbf{x}) = \underset{\mathbf{y}\in\mathrm{GEN}(\mathbf{x})}{\operatorname{argmax}}\left(\alpha \cdot \phi(\mathbf{y})\right)$$

Searching through the entire $\mathrm{GEN}(\mathbf{x})$ space is intractable even with a local model, so a beam-search algorithm is used. The search algorithm consumes one character input token at a time, and iterates through the existing beams to score two new alternative hypotheses by either appending the new character to the last word in the beam, or starting a new word at the current position.

Algorithm 1 Dual decomposition inference algorithm, and modified Viterbi and beam-search algorithms.

$\forall i \in \{1 \text{ to } |\mathbf{x}|\}: \ \forall k \in \{0,1\}: u_i(k) = 0$
for $t \leftarrow 1$ **to** T **do**
  $\mathbf{y}^{\mathbf{c}*} = \underset{\mathbf{y}}{\operatorname{argmax}}\ P(\mathbf{y}^c|\mathbf{x}) + \sum_{i\in|\mathbf{x}|} u_i(y_i^c)$
  $\mathbf{y}^{\mathbf{w}*} = \underset{\mathbf{y}\in\mathrm{GEN}(\mathbf{x})}{\operatorname{argmax}}\ F(\mathbf{y}^w|\mathbf{x}) - \sum_{j\in|\mathbf{x}|} u_j(y_j^w)$
  if $\mathbf{y}^{\mathbf{c}*} = \mathbf{y}^{\mathbf{w}*}$ **then**
    return $(\mathbf{y}^{c*}, \mathbf{y}^{w*})$
  end if
  for all $i \in \{1 \text{ to } |\mathbf{x}|\}$ **do**
    $\forall k \in \{0,1\} : u_i(k) = u_i(k) + \alpha_t(2k-1)(y_i^{w*} - y_i^{c*})$
  end for
end for
return $(\mathbf{y}^{\mathbf{c}*}, \mathbf{y}^{\mathbf{w}*})$

Viterbi:
$V_1(1) = 1, V_1(0) = 0$
for $i = 2$ to $|\mathbf{x}|$ **do**
  $\forall k \in \{0,1\} : V_i(k) = \underset{\mathbf{k}'}{\operatorname{argmax}} P_i(k|k')V_{i-1}k' + u_i(k)$
end for

Beam-Search:
for $i = 1$ to $|\mathbf{x}|$ **do**
  for item $v = \{w_0, \cdots, w_j\}$ in beam(i) **do**
    append x_i to w_j, score$(v) \stackrel{+}{=} u_i(0)$
    $v = \{w_0, \cdots, w_j, x_i\}$, score$(v) \stackrel{+}{=} u_i(1)$
  end for
end for

2.3 Combining Models with Dual Decomposition

Various mixing approaches have been proposed to combine the above two approaches (Wang et al., 2006; Lin, 2009; Sun et al., 2009; Sun, 2010; Wang et al., 2010). These mixing models perform well on standard datasets, but are not in wide use because of their high computational costs and difficulty of implementation.

Dual decomposition (DD) (Rush et al., 2010) offers an attractive framework for combining these

	Academia Sinica					Peking Univ.				
	R	P	F_1	R_{oov}	C	R	P	F_1	R_{oov}	C
Char-based CRF	95.2	93.6	94.4	58.9	0.064	94.6	95.3	94.9	77.8	0.089
Word-based Perceptron	95.8	**95.0**	**95.4**	**69.5**	0.060	94.1	95.5	94.8	76.7	0.099
Dual-decomp	**95.9**	94.9	**95.4**	67.7	**0.055**	**94.8**	**95.7**	**95.3**	**78.7**	**0.086**
	City Univ. of Hong Kong					Microsoft Research				
	R	P	F_1	R_{oov}	C	R	P	F_1	R_{oov}	C
Char-based CRF	94.7	94.0	94.3	**76.1**	0.065	96.4	96.6	96.5	71.3	0.074
Word-based Perceptron	94.3	94.0	94.2	71.7	0.073	97.0	97.2	97.1	74.6	0.063
Dual-decomp	**95.0**	**94.4**	**94.7**	75.3	**0.062**	**97.3**	**97.4**	**97.4**	**76.0**	**0.055**

Table 1: Results on SIGHAN 2005 datasets. R_{oov} denotes OOV recall, and *C* denotes segmentation consistency. Best number in each column is highlighted in bold.

two types of models without incurring high costs in model complexity (in contrast to (Sun et al., 2009)) or decoding efficiency (in contrast to bagging in (Wang et al., 2006; Sun, 2010)). DD has been successfully applied to similar situations for combining local with global models; for example, in dependency parsing (Koo et al., 2010), bilingual sequence tagging (Wang et al., 2013) and word alignment (DeNero and Macherey, 2011).

The idea is that jointly modelling both character-sequence and word information can be computationally challenging, so instead we can try to find outputs that the two models are most likely to agree on. Formally, the objective of DD is:

$$\max_{\mathbf{y}^c, \mathbf{y}^w} P(\mathbf{y}^c|\mathbf{x}) + F(\mathbf{y}^w|\mathbf{x}) \; s.t. \; \mathbf{y}^c = \mathbf{y}^w \quad (1)$$

where $\mathbf{y}^c$ is the output of character-based CRF, $\mathbf{y}^w$ is the output of word-based perceptron, and the agreements are expressed as constraints. *s.t.* is shorthand for "such that".

Solving this constrained optimization problem directly is difficult. Instead, we take the Lagrangian relaxation of this term as:

$$L(\mathbf{y}^c, \mathbf{y}^w, \mathbf{U}) = \quad (2)$$
$$P(\mathbf{y}^c|\mathbf{x}) + F(\mathbf{y}^w|\mathbf{x}) + \sum_{i \in |\mathbf{x}|} u_i(y_i^c - y_i^w)$$

where $\mathbf{U}$ is the set of Lagrangian multipliers that consists of a multiplier u_i at each word position i.

We can rewrite the original objective with the Lagrangian relaxation as:

$$\max_{\mathbf{y}^c, \mathbf{y}^w} \min_{\mathbf{U}} L(\mathbf{y}^c, \mathbf{y}^w, \mathbf{U}) \quad (3)$$

We can then form the dual of this problem by taking the $\min$ outside of the $\max$, which is an upper bound on the original problem. The dual form can then be decomposed into two sub-components (the two $\max$ problems in Eq. 4), each of which is local with respect to the set of Lagrangian multipliers:

$$\min_{\mathbf{U}} \Bigg(\max_{y^c} \left[P(\mathbf{y}^c|\mathbf{x}) + \sum_{i \in |\mathbf{x}|} u_i(y_i^c) \right] \quad (4)$$
$$+ \max_{y^w} \left[F(\mathbf{y}^w|\mathbf{x}) - \sum_{j \in |\mathbf{x}|} u_j(y_j^w) \right] \Bigg)$$

This method is called dual decomposition (DD) (Rush et al., 2010). Similar to previous work (Rush and Collins, 2012), we solve this DD problem by iteratively updating the sub-gradient as depicted in Algorithm 1.[1] In each iteration, if the best segmentations provided by the two models do not agree, then the two models will receive penalties for the decisions they made that differ from the other. This penalty exchange is similar to message passing, and as the penalty accumulates over iterations, the two models are pushed towards agreeing with each other. We also give an updated Viterbi decoding algorithm for CRF and a modified beam-search algorithm for perceptron in Algorithm 1. T is the maximum number of iterations before early stopping, and α_t is the learning rate at time t. We adopt a learning rate update rule from Koo et al. (2010) where α_t is defined as $\frac{1}{N}$, where N is the number of times we observed a consecutive dual value increase from iteration 1 to t.

[1] See Rush and Collins (2012) for a full introduction to DD.

3 Experiments

We conduct experiments on the SIGHAN 2003 (Sproat and Emerson, 2003) and 2005 (Emerson, 2005) bake-off datasets to evaluate the effectiveness of the proposed dual decomposition algorithm. We use the publicly available Stanford CRF segmenter (Tseng et al., 2005)[2] as our character-based baseline model, and reproduce the perceptron-based segmenter from Zhang and Clark (2007) as our word-based baseline model.

We adopted the development setting from (Zhang and Clark, 2007), and used CTB sections 1-270 for training and sections 400-931 for development in hyper-parameter setting; for all results given in tables, the models are trained and evaluated on the standard train/test split for the given dataset. The optimized hyper-parameters used are: ℓ_2 regularization parameter λ in CRF is set to 3; the perceptron is trained for 10 iterations with beam size 200; dual decomposition is run to max iteration of 100 (T in Algo. 1) with step size 0.1 (α_t in Algo. 1).

Beyond standard precision (P), recall (R) and F_1 scores, we also evaluate segmentation consistency as proposed by (Chang et al., 2008), who have shown that increased segmentation consistency is correlated with better machine translation performance. The consistency measure calculates the entropy of segmentation variations — the lower the score the better. We also report out-of-vocabulary recall (R_{oov}) as an estimation of the model's generalizability to previously unseen words.

4 Results

Table 1 shows our empirical results on SIGHAN 2005 dataset. Our dual decomposition method outperforms both the word-based and character-based baselines consistently across all four subsets in both F_1 and OOV recall (R_{oov}). Our method demonstrates a robustness across domains and segmentation standards regardless of which baseline model was stronger. Of particular note is DD's is much more robust in R_{oov}, where the two baselines swing a lot. This is an important property for downstream applications such as entity recognition. The DD algorithm is also more consistent, which would likely lead to improvements in applications such as machine translation (Chang et al., 2008).

The improvement over our word- and character-based baselines is also seen in our results on the earlier SIGHAN 2003 dataset. Table 2 puts our method in the context of earlier systems for CWS. Our method achieves the best reported score on 6 out of 7 datasets.

SIGHAN 2005				
	AS	PU	CU	MSR
Best 05	95.2	95.0	94.3	96.4
Zhang et al. 06	94.7	94.5	94.6	96.4
Z&C 07	94.6	94.5	95.1	97.2
Sun et al. 09	-	95.2	94.6	97.3
Sun 10	95.2	95.2	**95.6**	96.9
Dual-decomp	**95.4**	**95.3**	94.7	**97.4**
SIGHAN 2003				
Best 03	96.1	95.1	94.0	
Peng et al. 04	95.6	94.1	92.8	
Z&C 07	96.5	94.0	94.6	
Dual-decomp	**97.1**	**95.4**	**94.9**	

Table 2: Performance of dual decomposition in comparison to past published results on SIGHAN 2003 and 2005 datasets. Best reported F_1 score for each dataset is highlighted in bold. *Z&C 07* refers to Zhang and Clark (2007). *Best 03, 05* are results of the winning systems for each dataset in the respective shared tasks.

5 Discussion and Error Analysis

On the whole, dual decomposition produces state-of-the-art segmentations that are more accurate, more consistent, and more successful at inducing OOV words than the baseline systems that it combines. On the SIGHAN 2005 test set, in over 99.1% of cases the DD algorithm converged within 100 iterations, which gives an optimality guarantee. In 77.4% of the cases, DD converged in the first iteration. The number of iterations to convergence histogram is plotted in Figure 1.

Error analysis In many cases the relative confidence of each model means that dual decomposition is capable of using information from both sources to generate a series of correct segmentations better than either baseline model alone. The example below shows a difficult-to-segment proper name comprised of common characters, which results in undersegmentation by the character-based CRF and

[2] http://nlp.stanford.edu/software/segmenter.shtml

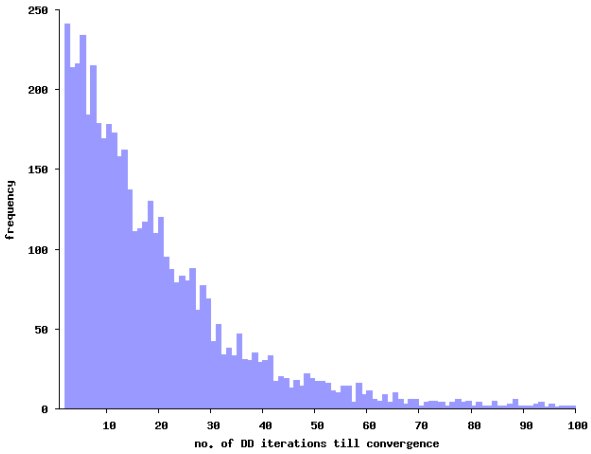

Figure 1: No. of iterations till DD convergence.

oversegmentation by the word-based perceptron, but our method achieves the correct middle ground.

Gloss	Tian Yage / 's / creations
Gold	田雅各 / 的 / 创作
CRF	田雅各的 / 创作
PCPT	田雅 / 各 / 的 / 创作
DD	田雅各 / 的 / 创作

A powerful feature of the dual decomposition approach is that it can generate correct segmentation decisions in cases where a voting or product-of-experts model could not, since joint decoding allows the sharing of information at decoding time. In the following example, both baseline models miss the contextually clear use of the word 点心 ("sweets / snack food") and instead attach 点 to the prior word to produce the otherwise common compound 一点点 ("a little bit"); dual decomposition allows the model to generate the correct segmentation.

Gloss	Enjoy / a bit of / snack food / , ...
Gold	享受 / 一点 / 点心 / ,
CRF	享受 / 一点点 / 心 / ,
PCPT	享受 / 一点点 / 心 / ,
DD	享受 / 一点 / 点心 / ,

We found more than 400 such surprisingly accurate instances in our dual decomposition output.

Finally, since dual decomposition is a method of joint decoding, it is still liable to reproduce errors made by the constituent systems.

6 Conclusion

In this paper we presented an approach to Chinese word segmentation using dual decomposition for system combination. We demonstrated that this method allows for joint decoding of existing CWS systems that is more accurate and consistent than either system alone, and further achieves the best performance reported to date on standard datasets for the task. Perhaps most importantly, our approach is straightforward to implement and does not require retraining of the underlying segmentation models used. This suggests its potential for broader applicability in real-world settings than existing approaches to combining character-based and word-based models for Chinese word segmentation.

References

Galen Andrew. 2006. A hybrid Markov/semi-Markov conditional random field for sequence segmentation. In *Proceedings of EMNLP*.

Ming-Hong Bai, Keh-Jiann Chen, and Jason S. Chang. 2008. Improving word alignment by adjusting chinese word segmentation. In *Proceedings of the third International Joint Conference on Natural Language Processing (IJCNLP)*.

Pichuan Chang, Michel Galley, and Chris Manning. 2008. Optimizing chinese word segmentation for machine translation performance. In *Proceedings of the ACL Workshop on Statistical Machine Translation*.

Keh-Jiann Chen and Shing-Huan Liu. 1992. Word identification for mandarin chinese sentences. In *Proceedings of COLING*.

Michael Collins. 2002. Discriminative training methods for hidden markov models: theory and experiments with perceptron algorithms. In *Proceedings of EMNLP*.

John DeNero and Klaus Macherey. 2011. Model-based aligner combination using dual decomposition. In *Proceedings of ACL*.

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

Jianfeng Gao, Mu Li, and Chang-Ning Huang. 2003. Improved source-channel models for Chinese word segmentation. In *Proceedings of ACL*.

Terry Koo, Alexander M. Rush, Michael Collins, Tommi Jaakkola, and David Sontag. 2010. Dual decomposition for parsing with non-projective head automata. In *Proceedings of EMNLP*.

Jonathan K. Kummerfeld, Daniel Tse, James R. Curran, and Dan Klein. 2013. An empirical examination of challenges in chinese parsing. In *Proceedings of ACL-Short*.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of 18th International Conference on Machine Learning (ICML)*.

Dekang Lin. 2009. Combining language modeling and discriminative classification for word segmentation. In *Proceedings of the 10th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing)*.

Alexander M. Rush and Michael Collins. 2012. A tutorial on dual decomposition and Lagrangian relaxation for inference in natural language processing. *JAIR*, 45:305–362.

Alexander M. Rush, David Sontag, Michael Collins, and Tommi Jaakkola. 2010. On dual decomposition and linear programming relaxations for natural language processing. In *Proceedings of EMNLP*.

Yanxin Shi and Mengqiu Wang. 2007. A dual-layer crfs based joint decoding method for cascaded segmentation and labeling tasks. In *Proceedings of Joint Conferences on Artificial Intelligence (IJCAI)*.

Richard Sproat and Thomas Emerson. 2003. The first international Chinese word segmentation bakeoff. In *Proceedings of the second SIGHAN workshop on Chinese language Processing*.

Xu Sun, Yaozhong Zhang, Takuya Matsuzaki, Yoshimasa Tsuruoka, and Jun'ichi Tsujii. 2009. A discriminative latent variable chinese segmenter with hybrid word/character information. In *Proceedings of HLT-NAACL*.

Weiwei Sun. 2010. Word-based and character-basedword segmentation models: Comparison and combination. In *Proceedings of COLING*.

Huihsin Tseng, Pichuan Chang, Galen Andrew, Daniel Jurasfky, and Christopher Manning. 2005. A conditional random field word segmenter for sighan bakeoff 2005. In *Proceedings of the fourth SIGHAN workshop on Chinese language Processing*.

Xinhao Wang, Xiaojun Lin, Dianhai Yu, Hao Tian, and Xihong Wu. 2006. Chinese word segmentation with maximum entropy and n-gram language model. In *Proceedings of the fifth SIGHAN workshop on Chinese language Processing*.

Kun Wang, Chengqing Zong, and Keh-Yih Su. 2010. A character-based joint model for chinese word segmentation. In *Proceedings of COLING*.

Mengqiu Wang, Wanxiang Che, and Christopher D. Manning. 2013. Joint word alignment and bilingual named entity recognition using dual decomposition. In *Proceedings of ACL*.

Nianwen Xue. 2003. Chinese word segmentation as character tagging. *International Journal of Computational Linguistics and Chinese Language Processing*, pages 29–48.

Yue Zhang and Stephen Clark. 2007. Chinese segmentation with a word-based perceptron algorithm. In *Proceedings of ACL*.

Ruiqiang Zhang, Genichiro Kikui, and Eiichiro Sumita. 2006. Subword-based tagging by conditional random fields for Chinese word segmentation. In *Proceedings of HLT-NAACL*.