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

Abstract

There are two dominant approaches to the Chinese word segmentation problem: 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 (**?**; **?**; **?**; **?**), 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") (**?**).

There are two primary classes of models: character-based (**?**; **?**; **?**; **?**) and word-based (**?**; **?**), with corresponding advantages and disadvantages. **?**) details their theoretical distinctions: 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. 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, it 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, and review existing approaches to combine these models.

2.1 Character-based Models

In the most commonly used contemporary approach to character-based segmentation, first proposed by (**?**), 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) (**?**) are widely adopted for this task, and give state-of-the-art results (**?**). In a first-order linear-chain CRF model, the conditional probability of a label sequence $\mathbf{y}$ given a word se-

quence $\mathbf{x}$ 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 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 (**?**). More recently, **?**) reported success using a linear model trained with the average perceptron algorithm (**?**). 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.

2.3 Combining Models with Dual Decomposition

Various mixing approaches have been proposed to combine the above two approaches (**?**; **?**; **?**; **?**; **?**). 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) (**?**) offers an attractive framework for combining these two types of models without incurring high costs in model complexity (in contrast to (**?**)) or decoding efficiency (in contrast to bagging in (**?**; **?**)). DD has been successfully applied to similar situations for combining local with global models; for example, in dependency parsing (**?**)), bilingual sequence tagging (**?**) and word alignment (**?**).

Algorithm 1 DD inference algorithm, modified Viterbi and beam-search.

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

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

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:

$$\max_{\mathbf{y}^c,\mathbf{y}^w}\ P(\mathbf{y}^c|\mathbf{x}) + F(\mathbf{y}^w|\mathbf{x}) \ \ni \mathbf{y}^c = \mathbf{y}^w$$

where $\mathbf{y}^c$ is the output of character-based CRF and $\mathbf{y}^w$ is the output of word-based perceptron.

The DD algorithm is an iterative procedure: 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

	AS					PU				
	R	P	F_1	R_{oov}	C_{onst}	R	P	F_1	R_{oov}	C_{onst}
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
	CU					MSR				
	R	P	F_1	R_{oov}	C_{onst}	R	P	F_1	R_{oov}	C_{onst}
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_{onst} denotes segmentation consistency. † and ‡ denote statistical significance ($p < 0.01$) against CRF and perceptron baselines, respectively.

towards agreeing with each other. We give an updated Viterbi decoding algorithm for CRF and a modified beam-search algorithm for perceptron, as well as pseudo-code for DD algorithm in Algo. 1[1].

3 Experiments

We conduct experiments on the SIGHAN 2003 (**?**) and 2005 (**?**) bake-off datasets to evaluate the effectiveness of the proposed dual decomposition algorithm. We use the publicly available Stanford CRF segmenter (**?**)[2] as our character-based baseline model, and reproduce the perceptron-based segmenter from **?**) as our word-based baseline model. We adopted the development setting from (**?**), and used CTB sections 1-270 for training, and sections 400-931for development. The optimized hyperparameters used are: ℓ_2 regularization parameter λ in CRF is set to 3; perceptron is trained for 10 iterations with beam size 200; dual decomposition is run to max iteration (T in Algo. 1) of 100 with step size (α_t in Algo. 1) 0.1. Other than the standard precision (P), recall (R) and F_1 scores, we also evaluate segmentation consistency as proposed by (**?**), 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. Statistical significance tests are done using the paired bootstrap resampling method (**?**).

[1]Due to space limitations, we defer to the tutorial of **?**) for a full introduction of DD.

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

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 significant improvement in R_{oov}, which is particularly important for downstream applications such as entity recognition. The DD algorithm is also more consistent (as measured by C_{onst}), which would likely to lead to improvements in applications such as machine translation (**?**).

The improvement over our word- and character-based baselines remains for 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.

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, over 99.1% cases the DD algorithm converged within 100 iterations which gives 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.

	AS	PU	CU	MSR
SIGHAN 2005				
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 **?**). *Best 03, 05* are results of the winning systems for each dataset in the respective shared tasks.

Error analysis Since dual decomposition is a method of joint decoding, it is liable to reproduce errors made by the constituent systems. In the example below, dual decomposition output follows the incorrect segmentation of the character-based CRF in oversegmenting the compound "sea, land, and air."

Gloss	Large-scale / sea, land, and air / joint / military exercises
Gold	大规模 / 海陆空 / 联合 / 军演
CRF	大规模 / 海 / 陆空 / 联合 / 军演
PCPT	大规模 / 海陆空 / 联合 / 军演
DD	大规模 / 海 / 陆空 / 联合 / 军演

Nevertheless, 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 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.

English	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.

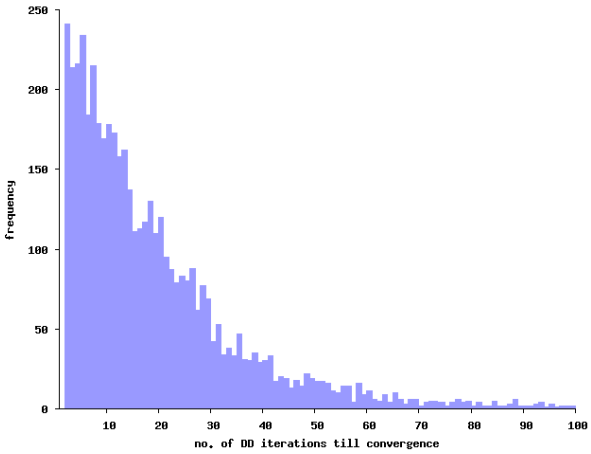

Figure 1: No. of iterations till DD convergence.

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.