Better Chinese Word Segmentation via Dual Decomposition (Title to be changed)

Author 1
XYZ Company
111 Anywhere Street
Mytown, NY 10000, USA
author1@xyz.org

Author 2
ABC University
900 Main Street
Ourcity, PQ, Canada A1A 1T2
author2@abc.ca

Abstract

There are two dominant approaches to Chinese word segmentation problem: word-based and character-based. They each have their own strengths and weaknesses. Word-based approaches are better at capturing longer context dependencies, and achieve higher segmentation consistency, whereas character-based approaches model the internal structures of words and can capture more out-of-vocabulary words. It is an intuitive idea to find methods to combine these approaches. Past efforts in combing word and character-based segmenters involve either training many instances of segmenters and polling their results, or designing complex latent-variable methods that incur heavy computational costs. In this paper, we propose an effective joint decoding method using dual decomposition, which does not require any additional training. Our method is simple and easy to implement, and achieves best reported results on 6 out of 7 standard SIGHAN evaluation 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 [cite, cite, cite - Pichuan, what else?], the quality and consistency of segmentation has important downstream impacts on system performance in machine translation, [and...].

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. Character-based segmenters better model the internal compositional structure of words and are therefore more effective at detecting out-of-vocabulary words; however, they suffer from relative inconsistency. Word-based segmenters are better at reproducing the words of the training lexicon and have a higher consistency, but are not as good at detecting OOV words. Word-based models have the additional advantage that they tend to capture information from significantly larger contextual spans.

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 on all 8 datasets. In particular, it improves OOV recall rates and consistency, and gives the best reported results to date on 6 out of 7 datasets.

2 Related Work

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 (CRFs) are often used for this purpose (cite, cite, cite). In a CRF segmentation model, the probability of a label sequence is given by this equation:

Common linguistic features include character n-grams and morphological suffix/prefix features. Since these features capture information about the compositional properties of characters, they are likely to generalize well to unknown words.

2.2 Word-based Models

Initially word-based segmentation approaches employed simple heuristics like dictionary-lookup maximum matching (?), contemporary approaches use machine learning techniques to solve an argmax problem of the form:

2.3 Mixing Models

Since the two types of models described above have different strengths and make different kinds of errors, various mixing approaches have been proposed to combine them (?; ?; ?; ?; ?).

(?) and (?) both try to incorporate a character-based local classifier model (MaxEnt) with a language model that captures more word-level context. In order to bring in bigram language model, (?) gave a heuristic decoding method that involves various conditioning and back-off; (?) gave a modified Viterbi algorithm actually has complexity of $O(T^3)$.

(?) is the most similar in that they also try to directly combine two segmenters – one char-based and one word-based. But they do it through segmenter bagging, which would require training up 50 or more individual segmenters and poll their results at test time, significantly increases training and testing effort, not practical. (?) presented a latent-variable model. very complicated, hard to train. our method is much simpler and works better than their method.

3 Dual-Decomposition

Dual decomposition offers a ideal framework for combining these two sources of signals without incurring high cost in model complexity (in contrast to (?)) or decoding efficiency (in contrast to bagging in (?; ?)). DD has been successfully applied to similar situations where we want to combine local model with global models, for example, in dependency parsing (?)), bilingual sequence tagging (?) and word alignment (). Give a brief description of DD algorithm, focus on the intuition. See (?) and (?) for a good short introduction example. Refer users to (?) for a full tutorial on dual decomp. The modification to Viterbi decoding is exactly the same as in (?) and (?). The modification to the beam-search is similar, each time we extend a hypothesis with a new character, depending if the new character is appended to the last word or starting a new word, the corresponding DD penalty is factored into the score for the new hypothesis.

4 Experiments

In this work, we employ two baseline models — a character-based CRF and a word-based perceptron — and test the performance of jointly decoding these baseline systems with dual decomposition. For our character-based CRF, we use the open-source Stanford CRF segmenter described in (?).[1] For our word-based perceptron, we use a reimplementation of (?).

For development data, we used: Training: CTB section 1-270 400-931 1001-115; Dev: CTB section 271-300.

Hyper-parameters tuned on the dev set. Perceptron: 10 iterations of training. CRF: L2-regularization Sigma set to 3. Dual-decomp: initial step size set to 0.1, 100 iterations.

4.1 Dataset

Give description of the SIGHAN 2003 (?) and 2005 (?) bake-off datasets. List a table of the corpora names, size of training, testing, etc, similar to (?).

5 Results

Reformat the table so it fits in a single column. The OOV rates should be moved out of this table and put

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

		#W.T.	#W	#C.T.	#C	OOV
2005	AS	-	-	-	-	4.3
	MS	88K	2.3M	5K	4.1M	2.6
	PU	55k	1.1M	5K	1.8M	5.8
	CU	69K	1.5M	5K	2.4M	7.4
2003	AS	-	-	-	-	2.2
	PU	-	-	-	-	6.9
	CU	-	-	-	-	7.1

Table 1: Results on SIGHAN 2005 and 2003 datasets.

in the data description tables in Experiment section Dataset subsection.

6 Discussion

6.1 Dual decomposition convergence

Plot the histogram of the number of iterations it takes to converge, and percentage of optimality.

6.2 Consistency

(**?**) have shown that increased segmentation consistency is correlated with better machine translation performance. Following their method for calculating the conditional entropy of a segmentation system, we see in Table [insert table] that our dual decomposition method achieves the most consistent results on 6 out of 7 datasets.

Consistency Results, Sighan 2003:

as crf 0.0428091404101 as pct 0.0441475626702 as dd 0.042168035925 cityu crf 0.0946373294134 cityu pct 0.10025983369 cityu dd 0.0867128300158 ctb crf 0.174004652166 ctb pct 0.18632727645 ctb dd 0.165767655651 pku crf 0.0423983535204 pku pct 0.0543269973345 pku dd 0.045000565176

Consistency Results, Sighan 2005:

as crf 0.0718479895225 as pct 0.0681887138263 as dd 0.0624082516494 cityu crf 0.0660880615807 cityu pct 0.0742624297 cityu dd 0.0627869624389 msr crf 0.0755273737469 msr pct 0.064051889613 msr dd 0.0558050161967 pku crf 0.0892311495453 pku pct 0.0998921318846 pku dd 0.0868198134889

6.3 Oracle

Following (**?**), we run an oracle experiment to estimate the upper bound of improvement possible via system combination to further contextualize our results.

SIGHAN 2005

		F_1	R_{oov}	R_{iv}
AS	CRF	94.4	58.9	96.9
	PCPT	95.4	69.5	97.0
	DD	95.4	67.7	97.2
	Best 05	95.2	69.6	96.3
	Zhang et al. 06	94.7	-	-
	Z&C 07	94.6	-	-
	Sun et al. 09	-	-	-
	Sun 10	95.2	-	-
PU	CRF 94.9	77.8	95.6	
	PCPT	94.8	76.7	95.2
	DD	95.3	78.7	95.8
	Best 05	95.0	63.6	97.2
	Zhang et al. 06	94.5	-	-
	Z&C 07	94.5	-	-
	Sun et al. 09	95.2	77.8	-
	Sun 10	95.2	-	-
CU	CRF	94.3	76.1	96.2
	PCPT	94.2	71.7	96.1
	DD	94.7	75.3	96.5
	Best 05	94.3	69.8	96.1
	Zhang et al. 06	94.6	-	-
	Z&C 07	95.1	-	-
	Sun et al. 09	94.6	68.8	-
	Sun 10	95.6	-	-
MS	CRF	96.5	71.3	97.1
	PCPT	97.1	74.6	97.6
	DD	97.4	76.0	97.9
	Best 05	96.4	71.7	96.8
	Zhang et al. 06	96.4	-	-
	Z&C 07	97.2	-	-
	Sun et al. 09	97.3	72.2	-
	Sun 10	96.9	-	-

SIGHAN 2003

AS	CRF	96.9	74.8	97.4
	PCPT	96.7	72.9	97.2
	DD	97.1	77.5	97.5
	Best 03	96.1	36.4	98.0
	Peng et al. 04	95.6	-	-
	Z&C 07	96.5	-	-
PU	CRF	95.3	80.3	96.5
	PCPT	95.0	79.0	96.0
	DD	95.4	80.6	96.5
	Best 03	95.1	72.4	97.9
	Peng et al. 04	94.1	-	-
	Z&C 07	94.0	-	-
CU	CRF	94.1	74.1	95.9
	PCPT	94.5	73.0	96.0
	DD	94.9	75.4	96.5
	Best 03	94.0	62.5	97.2
	Peng et al. 04	92.8	-	-
	Z&C 07	94.6	-	-

Table 2: Results on SIGHAN 2005 and 2003 datasets.

To do so, we combine our two baselines with the gold-standard segmentation. Each character in the test set is labeled with three *B* or *I* tags, *B* when it begins a word and *I* when it is word-medial or word-final, according to our two baselines and the gold standard. We then create oracle labels by majority vote: if the baselines agree, their label is used; if they disagree, the gold label is used.

The results of this oracle experiment, shown in Table [insert table], show that [what do they show? DD approaches the upper bound possible?]

Scoring Oracle Output, Sighan 2003 for as sighan2003/as.oracle 75 84 783 942 11989 11980 0.928 0.928 0.928 0.022 0.810 0.930 for cityu sighan2003/cityu.oracle 402 333 2364 3099 35087 35156 0.923 0.921 0.922 0.071 0.824 0.931 for ctb sighan2003/ctb.oracle 920 1005 4490 6415 39921 39836 0.862 0.864 0.863 0.181 0.773 0.882 for pku sighan2003/pku.oracle 186 187 1593 1966 17194 17193 0.896 0.897 0.897 0.069 0.834 0.901

Scoring Oracle Output, Sighan 2003 for as sighan2005/as.oracle 1855 726 9980 12561 122610 123739 0.913 0.904 0.908 0.043 0.722 0.921 for cityu sighan2005/cityu.oracle 577 371 3038 3986 40936 41142 0.917 0.912 0.914 0.074 0.817 0.925 for msr // currently broken sighan2005/msr.oracle 81 107 1066 1254 10987 10961 0.893 0.895 0.894 0.025 0.584 0.901 for pku sighan2005/pku.oracle 800 1661 9646 12107 104372 103511 0.892 0.899 0.895 0.058 0.815 0.896

6.4 Error analysis

Maybe look at cases where the model choose the worse of the two instead of the better of the two. See if there are patterns or insights we can draw. Not very important for a short paper.

some errors where DD picks the wrong model: gold: 有 亭台 流水 CRF: 亭台 ¡– DD PCT: 亭台

gold: 大模 海空 合 演 CRF: 大模 海 空 合 演 ¡– DD PCT: 大模 海空 合 演

gold: 在 李屋 附近 CRF: 在 李屋 附近 PCT: 在 李屋 附近 ¡– DD

gold: 父 笑 CRF: 父 笑 PCT: 父 笑 ¡– DD

gold: 天 冷 大 CRF: 天 冷 大 PCT: 天 冷 大 ¡– DD

7 Conclusion

In this paper we presented ... blah blah. Stress the simplicity and effectiveness of our method.