RC25411 (WAT1309-085) September 27, 2013
Computer Science

IBM Research Report

Efficient Domain-Adaptive Word Segmentation with Larger Context and Co-Training

Fei Huang, Abraham Ittycheriah, Salim Roukos
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598

Research Division
Almaden - Austin - Beijing - Cambridge - Haifa - India - T. J. Watson - Tokyo - Zurich

Efficient Domain-adaptive Word Segmentation with Larger Context and Co-training

Fei Huang huangfe@us.ibm.com
Abraham Ittycheriah abei@us.ibm.com
Salim Roukos roukos@us.ibm.com
IBM T.J. Watson Research Center

Abstract

Word segmentation is very important to many natural language processing tasks. In this paper we present a new word segmentation approach that efficiently and effectively combines word and character level larger context features with local features typically used in the CRF/MaxEnt models. We compare several feature and model combination strategies on multiple word segmentation tasks. Additionally, we propose a co-training method to train domain-specific LMs from unlabeled in-domain data. Thanks to the low cost of LM training (compared with the CRF/MaxEnt model training), it enables rapid development of domain-adaptive word segmenters with superior performance, achieving or outperforming state-of-the-art performances on several Chinese word segmentation tasks. When applying the improved segmenter to statistical machine translation, we observe consistent improvement in Bleu scores on multiple domain English-Chinese translation test sets.

1 Introduction

Many human languages (such as Chinese, Thai) have no explicit boundary marker between words. Accurate word segmentation is very important to many natural language processing tasks. Word segmentation errors will introduce out-of-vocabulary words, incorrect or inaccurate meaning of the sentence, and lead to errors in other language processing tasks, such as machine translation, information extraction and speech recognition. Table 1 shows translation errors from two state-of-the-art open MT systems due to incorrect word segmentation. As we can see, both systems do not recognize "奥巴马罗姆尼" as two person names ("Obama and Romney"): one system segments the string into "奥巴马罗姆" and "尼", and translates them into "Obama Roma"and "Nepal"respectively, while the other system transliterates the whole string to its pinyin form (the romanization of Chinese characters), "Aobamaluomuni".

Chinese Source	奥巴马罗姆尼决战"最后时刻".
English Reference	Obama and Romney fight "at the last minute".
MT System 1	*Obama Roma Nepal* decisive battle "last minute"
MT System 2	*Aobamaluomuni* Showdown "last moment"

Table 1: Translation errors due to incorrect word segmentation.

Word segmentation has been under study for a long time, and there are a lot of research on this topic. (Sproat and Shih, 2002) provides a detailed review. Here we select the following three categories of segmentation approaches for discussion and comparison.

(a) Rule-based segmentation

This approach segments words according to some rules either manually designed (such as maximum matching or minimum matching) or derived from manually segmented data (such as TBL-based segmentation (Palmer, 1997)). Taking the maximum matching as an example: with a predefined vocabulary, a stream of characters are segmented into words in the vocabulary. When multiple segmentations match the character sequence (e.g., "A/B/C", "A/BC", "ABC"), the one matching the longest word ("ABC") is chosen. Typically the vocabulary includes all

the characters from that language as single-character words so that any character sequence can be segmented. The drawbacks of this approach are: it requires the segmentation vocabulary, and it is unable to generate out-of-the-vocabulary (OOV) words. So a person name not within the vocabulary will be segmented incorrectly. More often, the name is segmented into a sequence of single characters.

(b) Finite state transducer (FST, aka FSM) based segmentation

This approach segments character sequence into a sequence of words so that the word sequence's n-gram language model (LM) cost is the smallest((Sproat et al., 1996) and (Lee et al., 2003)), where the word-based language model (an instance of finite state machine) is typically trained from manually segmented data. It only uses segmentation history (hypothesized words on the left of the current character) to compute the LM cost, and ignores richer context information (such as characters on the right) for segmentation.

(c) Feature-based segmentation

This approach exploits various contextual features (character, word, dictionary etc.) for segmentation, where the features and their weights are trained with maximum entropy (MaxEnt) models (Xue, 2003) or conditional random field (CRF) models (Peng et al., 2004) on manually segmented data. The Stanford segmenter (Tseng et al., 2005) adopting the CRF framework achieves the top performance in the 2005 Chinese word segmentation bakeoff, and is considered one of the state-of-the-art word segmenters (Chang et al., 2008). However, due to high computational cost in the CRF model training, limited number of features are utilized. As a result, only local context information is considered. Although this approach is able to generate new words not included in the predefined vocabulary or training data, it tends to under-segment and produces many OOVs.

In this paper we propose a new word segmentation approach. In particular:

1) It combines larger-context word and character-label features with local features used in the CRF/MaxEnt models. We explore different ways of combining long range history information with local context, and find that model combination at decoding time leads to the best segmentation performance.

2) It allows separate and iterative training of the language model with unsupervised learning. Because the LM training cost is significantly lower than that of the CRF model, it is much more efficient to select large amount of unlabeled, in-domain data to train domain-specific LMs. We propose a co-training approach, where the unlabeled data is automatically segmented by multiple segmenters. Combining different segmentations reinforces consistent segmentations (which are more likely correct). A new domain-specific LM is trained and incorporated in the new segmentation framework to further improve the segmentation performance.

3) We apply the proposed segmentation framework to Chinese word segmentation and observe significant performance improvement, achieving or outperforming state-of-the-art segmenters on both the 2005 Chinese word segmentation bakeoff test sets and in-house domain-specific segmentation tasks. Furthermore, when the improved segmenter is adopted for statistical machine translation (SMT), we observe consistent translation quality improvement across multiple domain test sets for English-Chinese translation.

In Section 2, we briefly describe the framework of the feature-based segmentation model. In section 3, we introduce larger context information for word segmentation, which is followed by three combination strategies in section 4. We discuss the co-training approach for domain-adaptive segmenter with unlabeled data in section 5. In section 6 we present experiment results of Chinese word segmentation on several test sets from multiple domains, followed by the experiments in machine translation. We conclude the paper with discussion in section 7.

2 Feature-based Word Segmentation

Conditional random field (CRF) is a statistical sequence modeling framework introduced by (Lafferty et al., 2001). (Peng et al., 2004) and (Tseng et al., 2005) applied this framework for Chinese word segmentation task. In this framework, word segmentation is cast as a sequence labeling problem: each character is labeled either as the first

character of a word (with label '**B**') or as the following character of a word (with label '**I**') [1]. Given a sequence of characters $X = x_1, \ldots x_T$ from a Chinese sentence, the model assigns a label to each character, and the probability of the label sequence $Y = y_1, \ldots y_T$ is

$$P_{crf}(Y|X) = \frac{1}{Z(X)} \exp(\sum_{t=1}^{T} \sum_{k} \lambda_k f_k(Y_t, X, t))$$

where t is the index of the current character in the character sequence, f_k is the k-th feature function (typically binary-valued) defined over the whole input sequence X and the current label sequence Y_t, and λ_k is the learned weight of function f_k. $Z(X)$ is the normalization term such that all the label sequences'probabilities sum to one. The feature weights are trained with a modified GIS algorithm (Darroch and Ratcliff, 1972) with adaptive step size so that the convergence is faster. The most probable label sequence can be determined using the Viterbi algorithm, which can also produce N-best list of labeling sequences.

$C_{-1}C_0C_1$	The central context features
$C_{-1}C_{-2}C_{-3}$	The left character context features
$C_1C_2C_3$	The right character context features
$L_{-1}L_{-2}$	The left label context features
C_mL_n	The conjunction of character and label features

Table 2: Feature types used in experiments: C_0 is the current character to be labeled, C_{-m}/C_m are the left/right m-th character, and L_n is the left n-th character label. Each context feature category also includes the substring features.

Typically, the feature functions are defined over the context of the current character, which includes the previous M characters, previous N character labels, the next P characters, as well as their conjunctions. After feature definition, the feature weights are learned from labeled data: sentences with words that are manually segmented. Due to the high cost of CRF model training, and the limited amount of labeled data, it is necessary to limit the total number of features to avoid possible data sparseness and slow training time. Therefore only local context information are incorporated in the feature engineering. In our setup, we set $M = 3, N = 2$ and $P = 3$. Table 2 shows the feature types used in our experiment.

The Maxent model is similar to the CRF framework in that it is also an exponential model, where the probability of a label given the observation is defined over a set of features, and these feature weights are learned from labeled data using GIS or SCGIS (Goodman, 2002). The CRF model considers the entire sequence information while the MaxEnt model makes the decision for each state (i.e., character) independently of other states, thus its training time is much faster than the CRF model training.

3 Larger Context Information for Word Segmentation

Like any sequence labeling problem, word segmentation benefits from more context information. In particular, the previous n-1 words are very powerful on predicting the current word. Such information is conveniently captured in a word-based n-gram LM . Considering that most Chinese words are composed of 2 characters, the model is able to capture history with roughly $2n$-1 characters. The information from longer history, combined with the local context information with rich feature types as used in the feature-based models, will be beneficial to the word segmentation problem.

3.1 Character-label n-gram features

The CRF/MaxEnt models are defined over the character sequence while a typical word-based LM is defined over the word sequence. To combine the two kind of information, one approach is to convert the word into characters. However, the character sequence does not encode the important word boundary information. Instead, we convert each word into a sequence of character-label (CharLabel) pairs. A history with n-1 words is mapped into a sequence of CharLabel pairs of length K, $\{(C_k, L_k)|k = 1, ..., K\}$, where K is the number of characters converted from the n-1 words, and the label L indicates whether the character C is the beginning of a word ('B') or not ('I'). These long range features are directly incor-

[1] These exists other labeling schemes such as "**S**"for single character word or "**E**"for the last character in a multi-character word.

porated into the CRF/MaxEnt framework for joint training, as discussed in section 4.1.

3.2 Character-label n-gram LM

The CharLabel features are binary features. To use real value information for more discriminative models, we build a character-label LM to compute the probability of any CharLabel sequence:

$$P_{clm}(C,L) = \prod_{i=1}^{K} p((c_i,l_i)|(c_{i-1},l_{i-1}),...,(c_{i-n+1},l_{i-n+1}))$$

where K is length of the sequence. The conditional joint probability is estimated based on the relative frequency of the CharLabel n-grams in the LM training data with modified Kneser-Ney smoothing (Chen and Goodman, 1999).

3.3 Word-based n-gram LM

Converting the word-based LM into CharLabel LM could lose some information about word history. Given a CharLabel token belonging to different words, such as "(本, I)"in "日本/Japan"and "书本/book", the two words end up with the same history when "(本, I)"is the last token in the LM history. To avoid such information loss, we also introduce the word-based LM.

Compared with the CRF and MaxEnt model training cost, training a n-gram LM is much more efficient. It is feasible to train the LM separately on larger amount of automatically segmented data, in addition to the limited amount of manually labeled data. In our experiment, we built a 7-gram character-label LM and 5-gram word-based LM, using them to score the partial hypotheses at decoding time (for model interpolation) or full hypotheses (for N-best rescoring), as discussed in the following section.

4 Combining Larger Context Information with Local Features

We explore different ways of combining larger context information with the CRF/MaxEnt models. The combination could be applied at training time as feature combination, at decoding time as model combination, or after decoding as hypothesis re-ranking.

4.1 Feature Combination

Both the CRF or MaxEnt framework allow the combination of long range character-label sequence with existing local features in one framework. Training all the feature weights together enables jointly maximizing the conditional probability with the interaction between features from both families. The drawback is, when the history length (n) is large, the number of n-gram features is huge. In our experiment, 56K manually segmented sentences (1.6M words) produce nearly 10M 7-gram character-label features. It takes long time to train a CRF model with this huge feature set, combined with the original local context features (~14M features). Instead we train a MaxEnt model combining both family of features.

4.2 Model Combination

To avoid the expensive training cost with huge amount of features, we combine the LMs trained from larger context with the CRF/MaxEnt models trained with local features. Given a sequence of characters X, we want to find the label sequence Y^* such that

$$Y^* = \arg\max(\lambda_{fm}C_{fm}(Y,X) + \lambda_{lm}C_{lm}(Y,X))$$

where C_{fm} is the cost of the feature-based model, and C_{lm} is the cost of the n-gram language model. Taking the CRF model as an example,

$$C_{fm}(Y,X) = \log P_{crf}(Y|X)$$
$$C_{lm}(Y,X) = \lambda_{clm}C_{clm}(Y,X) + \lambda_{wlm}C_{wlm}(Y,X)$$

where

$$C_{clm}(Y,X) = \log P_{clm}(Y|X)$$
$$C_{wlm}(Y,X) = \log P_{wlm}(Y|X)$$

$P_{crf}(Y|X)$ is the probability of the label sequence given the character sequence computed by the CRF model; $P_{clm}(Y,X)$ is the LM probability of the CharLabel sequence, and $P_{wlm}(Y,X)$ is the word-based LM probability of the word sequence (converted from the CharLabel sequence). λ_{fm}, λ_{lm}, λ_{clm} and λ_{wlm} are the models'interpolation weights, which are empirically chosen with the segmentation development set. In our experiment, we set $\lambda_{fm} = 1.0$, $\lambda_{lm} = 0.8$, $\lambda_{clm} = 1.0$ and $\lambda_{wlm} = 0.05$.

During decoding time, the Viterbi inference algorithm is used to find the optimal label sequence. At each state we need compute both the CRF model cost and the word and CharLabel LM costs, therefore we also keep track of partial segmentation hypothesis, i.e., the previous n character

and their labels, as the LM history. The weighted sum of the costs is the overall cost for each node. Search with larger context is a little more complicated than using the stand-alone CRF model, it is still manageable given that the history length is not too long.

4.3 LM-based Hypothesis Re-scoring

The LM information can also be used for N-best hypothesis rescoring after decoding. Both the CRF and MaxEnt segmenters are able to generate top-N label sequences, which are converted word streams. We compute the sentence level perplexity for each hypothesis with the word LM, then select the one with the smallest perplexity as the final label sequence.

Such approach can be compared with the combination of LM for discriminative word segmentation as proposed in (Lin, 2009), where the segmented words from first-pass discriminative segmentation are re-grouped in the second-pass word-based LM re-scoring. As the decoding and re-scoring are two separate steps, it is quite possible that the first pass segmentation errors are introduced into the second pass re-grouping and not recoverable. Our experiments also show that model combination at decoding time achieves better performance than the hypothesis re-scoring strategy.

We compare the three combination strategies: feature combination, model combination and hypothesis re-scoring. Their performance is shown in section 6.1.

5 Co-training for Domain-adaptive Segmenter

Another advantage of training the CRF/MaxEnt model and the LM separately is that LM can be trained with a lot more data thanks to its lower training cost. These data are not necessarily manually segmented. Instead, they can be selected from specific target domains, and automatically segmented using existing segmenters.

One strategy is based on self-training (Yarowsky, 1995): the unlabeled data is first segmented with a segmenter, then the segmenter is re-trained from these automatically segmented data. The disadvantage is that errors from the first pass segmentation are included as the segmenter's training data, which could lead to more errors (Wang et al., 2011).

Here we propose a co-training (Blum and Mitchell, 1998) based strategy: we select two baseline segmenters (FSM and CRF segmenter), use them to segment the unlabeled data. If they produce the same segmentation for a sentence, we assume it is correct. We concatenate the two segmentations, thus increase the weights of consistent segmentations, then re-train the word and Char-Label LMs. The newly trained LMs are incorporated in the CRF framework for model combination. The LM trained from such data is able to improve the segmentation accuracy for the target domain when combined with the feature-based segmenter. Such co-training approach encourages the segmenters to be trained with consistent in-domain segmentations (which are more likely correct), while allows different segmentations to be selected according to the LM probabilities. The improved segmenter can be applied on the same unlabeled data for better LM training and word segmentation. Such iterative co-training process allows the training of domain-specific word segmenter when the unlabeled data are selected from the target domains.

6 Word Segmentation Evaluation

6.1 Multi-domain Chinese Word Segmentation

Our Chinese word segmentation training data include 56K sentences, which corresponds to 1.6M manually segmented words. These data are from the LDC Chinese Treebank news domain data (CTB-7) and patent domain data. We first trained a standard MaxEnt word segmenter using the features introduced in section 2, then we experimented with different strategies of combining larger context features. Our test set is from online discussion forum (DF), a genre under investigation in the BOLT project. The segmentation results are shown in Table 3. The larger context features improve the MaxEnt segmenter using all the three combination methods. However, feature combination within the MaxEnt model brings smaller improvement than training separate LMs for hypothesis re-scoring and model combination, not to mention that the cost of training LMs is much less. Using the LM for model combination at decoding time outperforms hypothesis rescoring, because the first option allows hypothesis search in a larger space. For the following experiment, we only used the model combination strategy to combine the MaxEnt/CRF models and LM models.

Model	F-score
MaxEnt	89.5
MaxEnt + n-gram features (feature combination)	89.8
MaxEnt +LM (model combination)	94.0
MaxEnt + LM (hyp rescoring)	90.9

Table 3: Different ways of combining n-gram features within the MaxEnt framework.

In the following experiment, we compared several segmenters on multiple test sets. These segmenters include a standard CRF segmenter, a CRF+LM segmenter (using model combination) and an FSM-based word segmenter, all trained with the same manually segmented data. The test sets include a news domain test set selected from the Chinese Treebank (CTB devset), a patent domain test set, as well as the abovementioned online discussion forum (DF) test set. Additionally, we also compare with the Stanford word segmenter, which achieved several top scores in the 2005 Chinese Word Segmentation bakeoff.

F-score	CTB	DF	Patent
number of words	37166	58671	13800
FSM	93.4	90.9	94.3
Stanford Segmenter	N/A[2]	93.2	88.4
CRF	95.4	90.9	94.2
CRF+LM Sup56K	96.6	93.7	95.2

Table 4: Word segmentation performance (F-score) of the baseline CRF model and the proposed CRF+LM model on three test sets from different domains and genres.

As seen in Table 4, when the test set matches the domain of the training data, the CRF segmenter performs significantly better than the FSM segmenter (95.4 vs. 93.4 on the Chinese Treebank test set) because the learned model fits the test domain very well. The CRF segmenter performs much better on the patent domain test set than the Stanford segmenter (94.2 vs. 88.4), and worse on the discussion forum test set (90.9 vs. 93.2). This is because the Stanford segmenter includes some dictionary resources that the CRF segmenter did not use, while the CRF segmenter is trained with patent domain data which are not available to the Stanford segmenter. The LM trained with 56K manually segmented sentences (CRF+LM Sup56K) brings 1.0-2.8 pt gain to the CRF segmenter across all three test sets, which demonstrates the effectiveness of the larger context information from LM.

F-score	CTB	DF	Patent
Sup56K	96.6	93.7	95.2
Unsup Self-training	97.1	94.2	95.7
Unsup Co-training	97.3	94.3	96.1

Table 5: Word segmentation performance (F-score) of supervised and unsupervised training: self-training vs. co-training.

We evaluated the effectiveness of unsupervised learning, and the results are shown in Table 5. We first adopted the self-training strategy. We selected additional 2M unsegmented sentences from the discussion forum domain, and 1.4M sentences from the patent domain, automatically segmented them into words with the CRF+LM (Sup56K) segmenters, respectively. We trained domain-specific language models with the combination of manually and automatically segmented in-domain data. This new LM from the unsupervised learning yields additional 0.5 pt gain in F-score. We also experimented with the co-training strategy and observed additional 0.2-0.4 pt improvement over the self-training. This is because higher weights are given to the consistently segmented in-domain data, which makes the re-trained model more robust to segmentation errors from any single segmenter.

This result is encouraging: it demonstrates the feasibility of rapid domain adaptation for word segmentation. The CRF model training is computationally more expensive, thus it is preferable to train the CRF model on high quality, relatively small amount of manually segmented data. On the other hand, LM training is easy and fast, therefore it is possible to train a domain-specific LM from

[2]The Stanford segmenter is trained with the Chinese Treebank data. As the training data overlaps with our test data, we do not evaluate the Stanford Segmenter on this test set.

large amount of unlabeled in-domain data. Combining both models enables more accurate model training and rapid domain adaptation with unlabeled data.

F-score	Training time	w/o LM	w/ LM
MaxEnt	1hr	89.5	94.0
CRF	12hr	90.9	94.2

Table 6: MaxEnt and CRF word segmenter on the discussion forum test set, with and without the LM cost.

We also compared the training time of the MaxEnt model and CRF model for word segmentation in Table 6. The MaxEnt model is trained with SCGIS (Goodman, 2002), whose training time is a magnitude faster than the CRF model training, with lower performance compared with the CRF segmenter, as seen in Table 6 (89.5 vs. 90.9). However, when combining the LM model with the MaxEnt model, the gap between the two segmenters is much smaller, with significantly reduced model training time. It shows another advantage of model combination: speed up model training with little performance degradation.

6.2 Chinese Word Segmentation Bakeoff Task

This tasks is from the second international Chinese word segmentation bakeoff held by SIGHAN in 2005 to evaluate the state -of -the -art in Chinese word segmentation. In this task, corpora from four organizations are provided for word segmentation training and test. These corpora represent Chinese text from different encodings (Simplified Chinese and Traditional Chinese), different regions (Mainland China, Hong Kong SAR and Taiwan) and the segmentation truth follows different segmentation guidelines. Table 7 lists the corpora statistics. Details about the word segmentation bakeoff can be found at `http://www.sighan.org/bakeoff2005/`.

For each corpus, we first trained a CRF-based word segmenter from the training data. Additionally, we converted the word-based training data into character-label sequence and train a character-label n-gram LM, which is combined with the CRF model for segmentation. We also trained a word-based LM, whose cost is combined with CharLabel LM through interpolation. The interpolation weights are obtained from the devset performance selected from the Chinese Treebank. We compare the three segmentation models: the baseline CRF model (CRF), the CRF+CharLabel LM model (+CharLabel), and the above model with an additional word-based LM (++Word). We also compared them with the best segmenters from the 2005 word segmentation bakeoff on the closed test track (Best-2005) , where no resource other than the provided training corpora is allowed to train the segmenter. We also listed the current state-of-the-art performance on those test set from (Wang et al., 2012) (Wang2012).

Corpora	Encoding	Training Words	Test Words
Academia Sinica (AS)	Traditional Chinese	5,499,581	122,610
City University of Hong Kong (CityU)	Traditional Chinese	1,455,630	40,936
Microsoft Research China (MSR)	Simplified Chinese	2,368,391	106,873
Peiking University (PKU)	Simplified Chinese	1,109,947	104,372

Table 7: Word segmentation bakeoff corpora statistics.

Corpora	AS	CityU	MSR	PKU
CRF	93.8	92.8	94.2	93.8
+CharLabel	95.6	95.4	96.8	95.4
++Word	**95.7**	**95.4**	**97.0**	**95.7**
Best-2005	95.2	94.3	96.4	95.0
Wang-2012	95.6	95.6	97.2	95.7

Table 8: Word segmentation performance (F-score) of the baseline CRF model and the proposed CRF+LM (char and word) models in the word segmentation bakeoff test set.

The performances of these systems are reported in Table 8. We observed significant improvements (up to 2.2 pts) with the new combined model consistently in all four test sets. In particular, the

CharLabel LM leads to 2.1 F-score gain on average and the word-based LM brings additional 0.15 pt gain. The proposed model outperforms the best segmenter from the 2005 Bakeoff in all 4 closed test sets and achieves current state-of-the-art performance. In the AS test set, it even slightly better than the performance of the best open system (95.6), where any other material including material from other training corpora, proprietary dictionaries, internet and so forth are allowed to use for the segmentation. To the author's best knowledge, this is the highest performance reported on this test set.

6.3 Machine Translation Evaluation

In addition to evaluating on several word segmentation tasks, we also compare the baseline FSM segmenter, CRF segmenter and the improved CRF+LM Chinese word segmenter on English-Chinese machine translation task.

The English-Chinese translation system is trained with 20M sentence pairs, with roughly 10M sentences from LDC released data (covering various news domains and UN data) and 10M sentence pairs about software manual translations. The Chinese data are segmented with selected word segmenters. After data preprocessing (tokenization, word segmentation etc.), we run HMM and MaxEnt (Ittycheriah and Roukos, 2005) word alignment, and extract phrase translation pairs and translation rules from the aligned parallel data. The decoder is a chart-based decoder similar to the system described in (Zhao and Al-Onaizan, 2008). We used phrase translation pairs, Hiero-style synchronous context-free grammar (Chiang, 2005) and syntax tree-to-string rules for the translation, with feature cost functions capturing the rule-level, block-level and word-level translations as well as distortion models, sentence length models. The system is tuned with PRO tuning (Hopkins and May, 2011) on the NIST MT08 English-Chinese test set. We choose this as the tuning set because it has 4 reference translations with high translation quality. When computing the BLEU score (BLEU-4) (Papineni et al., 2002) for English-Chinese MT, the reference translation and the MT hypothesis are converted into characters in order to eliminate the variance introduced by word segmentation. In the MT08 open evaluation the best result on this test is 41.42 (constrained training track), which is similar to our baseline result using the FSM word segmenter. The test set includes a blog-style discussion forum test data from BOLT project (the same domain as in the word segmentation DF test set) and an eSupport test data (which is about online technical support and software manual translation). Both test sets have only single reference translation. Table 9 shows the character BLEU-4 scores of the translation results when the Chinese text is segmented with the FSM segmenter, the CRF segmenter, or the CRF+LM segmenter. Everything else remaining the same, the improved CRF+LM segmenter leads to 1.5-2.5 Bleu point improvement over the FSM-segmented MT system, and 0.6-1.3 pt gain over the CRF-segmented MT system.

It has been noticed that better word segmentation does not guarantee improved translation performance. For example, when a Chinese word is consistently over-segmented into multiple single character words, its translations can still be captured in phrase translation pairs. In our experiments, we observed consistent improvement on translation quality with multiple test sets. The improvement is particularly obvious on technical domain test set, where correct segmentation of domain-specific terms enables better word alignment quality, which in turn results in better phrase translation pairs and translation rules.

Bleu	MT08	eSupport	DF
# of sentences	1859	600	1844
FSM	41.32	30.18	16.18
CRF	41.40	31.52	17.11
CRF+LM	42.72	32.64	17.72

Table 9: English-Chinese translation improvement with different segmentations. The MT08 tuning set has 4 references while the other two test sets only have single reference.

7 Discussion and Conclusion

We presented a new word segmentation approach that efficiently and effectively combines larger context features (CharLabel and word-based LMs) with CRF/MaxEnt models that only use local context features. We compare several combination strategies: combined feature training within the CRF/MaxEnt framework, interpolation-based model combination at decoding time and hypothesis re-scoring after decoding. We find that model

combination at decoding time obtains the best result.

By iteratively updating the character-label and word LMs on labeled and unlabeled data through co-training, this approach enables rapid development of domain-specific word segmenter with superior performance, outperforming the CRF/MaxEnt word segmenter on several Chinese word segmentation tasks. When applying the improved segmenter to statistical machine translation, we observed consistent improvement in Bleu scores (up to 2.5 pts) for English-Chinese translations in multiple domain test sets.

Combining generative models with discriminative models for word segmentation has been proposed in (Lin, 2009) and (Wang et al., 2012). In Lin's work, a word-based LM is used for second pass word regrouping after the first-pass segmentation with an MaxEnt model. From our experiment, we find that integrated model combination achieves better result since both models play important roles in the label decision process. The CharLabel and word LMs can be naturally combined with the MaxEnt/CRF framework to improve the segmentation accuracy. As for using unsupervised data to build LM for improved word segmentation, Lin's work used 5G and 50G news corpus with $\tilde{0}$.4% F-score gains, while in our proposed model we obtained 0.5-2% gains with 2M sentences (about 40-60M words). The better performance is largely due to the tight combination of the LM and MaxEnt/CRF segmentation models during decoding. (Wang et al., 2012) combined the character-based LM with the CRF model, but it only exploited local context information (3-gram) and missed useful information from larger context and word-based LM. We also proposed co-training strategy for word segmentation, that achieves better results than the self-training strategy for domain-adaptive word segmentation. In addition to the word segmentation improvement, we also demonstrated improvement in English-Chinese MT, especially for domain-specific MT test sets.

The above model combination framework can be applied to other sequence labeling tasks. We have applied the similar technique to the mention detection task and observed promising preliminary results (0.5F improvement over a state-of-the-art IE system). We will continue in this direction and explore other ways to improve the combination strategy.

References

Avrim Blum and Tom Mitchell. 1998. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, COLT' 98, pages 92–100, New York, NY, USA. ACM.

Pi-Chuan Chang, Michel Galley, and Christopher D. Manning. 2008. Optimizing Chinese word segmentation for machine translation performance. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 224–232, Columbus, Ohio, June. Association for Computational Linguistics.

Stanley F Chen and Joshua Goodman. 1999. An empirical study of smoothing techniques for language modeling. *Computer Speech & Language*, 13(4):359–393.

David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *In ACL*, pages 263–270.

John N Darroch and Douglas Ratcliff. 1972. Generalized iterative scaling for log-linear models. *The annals of mathematical statistics*, 43(5):1470–1480.

Joshua Goodman. 2002. Sequential conditional generalized iterative scaling. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 9–16. Association for Computational Linguistics.

Mark Hopkins and Jonathan May. 2011. Tuning as ranking. In *In Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*.

Abraham Ittycheriah and Salim Roukos. 2005. A maximum entropy word aligner for arabic-english machine translation. In *In Proceedings of HLT-EMNLP*, pages 89–96.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

Young-Suk Lee, Kishore Papineni, Salim Roukos, Ossama Emam, and Hany Hassan. 2003. Language model based arabic word segmentation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 399–406. Association for Computational Linguistics.

Dekang Lin. 2009. Combining language modeling and discriminative classification for word segmentation. In *Computational Linguistics and Intelligent Text Processing*, pages 170–182. Springer.

David D Palmer. 1997. A trainable rule-based algorithm for word segmentation. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*, pages 321–328. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Fuchun Peng, Fangfang Feng, and Andrew McCallum. 2004. Chinese segmentation and new word detection using conditional random fields. In *Proceedings of the 20th international conference on Computational Linguistics*, page 562. Association for Computational Linguistics.

Richard Sproat and Chilin Shih. 2002. Corpus-based methods in chinese morphology and phonology. *COOLING 2002*.

Richard Sproat, William Gale, Chilin Shih, and Nancy Chang. 1996. A stochastic finite-state word-segmentation algorithm for chinese. *Computational linguistics*, 22(3):377–404.

Huihsin Tseng, Pichuan Chang, Galen Andrew, Daniel Jurafsky, 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*, volume 171. Jeju Island, Korea.

Yiou Wang, Jun'ichi Kazama, Yoshimasa Tsuruoka, Wenliang Chen, Yujie Zhang, and Kentaro Torisawa. 2011. Improving chinese word segmentation and pos tagging with semi-supervised methods using large auto-analyzed data. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 309–317.

Kun Wang, Chengqing Zong, and Keh-Yih Su. 2012. Integrating generative and discriminative character-based models for chinese word segmentation. *ACM Transactions on Asian Language Information Processing (TALIP)*, 11(2):7.

Nianwen Xue. 2003. Chinese word segmentation as character tagging. *Computational Linguistics and Chinese Language Processing*, 8(1):29–48.

David Yarowsky. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd annual meeting on Association for Computational Linguistics*, ACL '95, pages 189–196, Stroudsburg, PA, USA. Association for Computational Linguistics.

Bing Zhao and Yaser Al-Onaizan. 2008. Generalizing local and non-local word-reordering patterns for syntax-based machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 572–581. Association for Computational Linguistics.