Optimizing Chinese Word Segmentation for Machine Translation Performance

Pi-Chuan Chang, **Michel Galley**, and **Christopher D. Manning**
Computer Science Department, Stanford University
Stanford, CA 94305
pichuan,galley,manning@cs.stanford.edu

Abstract

Previous work has shown that Chinese word segmentation is useful for machine translation to English, yet the way different segmentation strategies affect MT is still poorly understood. In this paper, we demonstrate that optimizing segmentation for an existing segmentation standard does not always yield better MT performance. We find that other factors such as segmentation consistency and granularity of Chinese "words" can be more important for machine translation. Based on these findings, we implement methods inside a conditional random field segmenter that directly optimize segmentation granularity with respect to the MT task, providing an improvement of 0.73 BLEU. We also show that improving segmentation consistency using external lexicon and proper noun features yields a 0.32 BLEU increase.

1 Introduction

Word segmentation is considered an important first step for Chinese natural language processing tasks, because Chinese words can be composed of multiple characters but with no space appearing between words. Almost all tasks could be expected to benefit by treating the character sequence "天花" together, with the meaning *smallpox,* rather than dealing with the individual characters "天" (*sky*) and "花" (*flower*). Without a standardized notion of a word, traditionally, the task of Chinese word segmentation starts from designing a segmentation standard based on linguistic and task intuitions, and then aiming to building segmenters that output words that conform to the standard. One widely used standard is the Penn Chinese Treebank (CTB) Segmentation Standard (Xue et al., 2005).

It has been recognized that different NLP applications have different needs for segmentation. Chinese information retrieval (IR) systems benefit from a segmentation that breaks compound words into shorter "words" (Peng et al., 2002), paralleling the IR gains from compound splitting in languages like German (Hollink et al., 2004), whereas automatic speech recognition (ASR) systems prefer having longer words in the speech lexicon (Gao et al., 2005). However, despite a decade of very intense work on Chinese to English machine translation (MT), the way in which Chinese word segmentation affects MT performance is very poorly understood. With current statistical phrase-based MT systems, one might hypothesize that segmenting into small chunks, including perhaps even working with individual characters would be optimal. This is because the role of a phrase table is to build domain and application appropriate larger chunks that are semantically coherent in the translation process. For example, even if the word for *smallpox* is treated as two one-character words, they can still appear in a phrase like "天 花→*smallpox*", so that *smallpox* will still be a candidate translation when the system translates "天" "花". Nevertheless, Xu et al. (2004) show that an MT system with a word segmenter outperforms a system working with individual characters in an alignment template approach. On different language pairs, (Koehn and Knight, 2003) and (Habash and Sadat, 2006) showed that data-driven methods for splitting and preprocessing can improve Arabic-English and German-English MT.

Beyond this, there has been no finer-grained analysis of what style and size of word segmentation is optimal for MT. Moreover, most discussion of segmentation for other tasks relates to the size units to identify in the segmentation standard: whether to join or split noun compounds, for instance. People

Proceedings of the Third Workshop on Statistical Machine Translation, pages 224–232,
Columbus, June 2008.

generally assume that improvements in a system's word segmentation accuracy will be monotonically reflected in overall system performance. This is the assumption that justifies the concerted recent work on the independent task of Chinese word segmentation evaluation at SIGHAN and other venues. However, we show that this assumption is false: aspects of segmenters other than error rate are more critical to their performance when embedded in an MT system. Unless these issues are attended to, simple baseline segmenters can be more effective inside an MT system than more complex machine learning based models, with much lower word segmentation error rate.

In this paper, we show that even having a basic word segmenter helps MT performance, and we analyze why building an MT system over individual characters doesn't function as well. Based on an analysis of baseline MT results, we pin down four issues of word segmentation that can be improved to get better MT performance. *(i)* While a feature-based segmenter, like a support vector machine or conditional random field (CRF) model, may have very good aggregate performance, inconsistent context-specific segmentation decisions can be quite harmful to MT system performance. *(ii)* A perceived strength of feature-based systems is that they can generate out-of-vocabulary (OOV) words, but these can hurt MT performance, when they could have been split into subparts from which the meaning of the whole can be roughly compositionally derived. *(iii)* Conversely, splitting OOV words into noncompositional subparts can be very harmful to an MT system: it is better to produce such OOV items than to split them into unrelated character sequences that are known to the system. One big source of such OOV words is named entities. *(iv)* Since the optimal granularity of words for phrase-based MT is unknown, we can benefit from a model which provides a knob for adjusting average word size.

We build several different models to address these issues and to improve segmentation for the benefit of MT. First, we emphasize lexicon-based features in a feature-based sequence classifier to deal with segmentation inconsistency and over-generating OOV words. Having lexicon-based features reduced the MT training lexicon by 29.5%, reduced the MT test data OOV rate by 34.1%, and led to a 0.38 BLEU point gain on the test data (MT05). Second, we extend the CRF label set of our CRF segmenter to identify proper nouns. This gives 3.3% relative improvement on the OOV recall rate, and a 0.32 improvement in BLEU. Finally, we tune the CRF model to generate shorter or longer words to directly optimize the performance of MT. For MT, we found that it is preferred to have words slightly shorter than the CTB standard.

The paper is organized as follows: we describe the experimental settings for the segmentation task and the task in Section 2. In Section 3.1 we demonstrate that it is helpful to have word segmenters for MT, but that segmentation performance does not directly correlate with MT performance. We analyze what characteristics of word segmenters most affect MT performance in Section 3.2. In Section 4 and 5 we describe how we tune a CRF model to fit the "word" granularity and also how we incorporate external lexicon and information about named entities for better MT performance.

2 Experimental Setting

2.1 Chinese Word Segmentation

For directly evaluating segmentation performance, we train each segmenter with the SIGHAN Bakeoff 2006 training data (the UPUC data set) and then evaluate on the test data. The training data contains 509K words, and the test data has 155K words. The percentage of words in the test data that are unseen in the training data is 8.8%. Detail of the Bakeoff data sets is in (Levow, 2006). To understand how each segmenter learns about OOV words, we will report the F measure, the in-vocabulary (IV) recall rate as well as OOV recall rate of each segmenter.

2.2 Phrase-based Chinese-to-English MT

The MT system used in this paper is Moses, a state-of-the-art phrase-based system (Koehn et al., 2003). We build phrase translations by first acquiring bidirectional GIZA++ (Och and Ney, 2003) alignments, and using Moses' grow-diag alignment symmetrization heuristic.[1] We set the maximum phrase length to a large value (10), because some segmenters described later in this paper will result in shorter

[1]In our experiments, this heuristic consistently performed better than the default, grow-diag-final.

words, therefore it is more comparable if we increase the maximum phrase length. During decoding, we incorporate the standard eight feature functions of Moses as well as the lexicalized reordering model. We tuned the parameters of these features with Minimum Error Rate Training (MERT) (Och, 2003) on the NIST MT03 Evaluation data set (919 sentences), and then test the MT performance on NIST MT03 and MT05 Evaluation data (878 and 1082 sentences, respectively). We report the MT performance using the original BLEU metric (Papineni et al., 2001). All BLEU scores in this paper are uncased.

The MT training data was subsampled from GALE Year 2 training data using a collection of character 5-grams and smaller *n*-grams drawn from all segmentations of the test data. Since the MT training data is subsampled with character *n*-grams, it is not biased towards any particular word segmentation. The MT training data contains 1,140,693 sentence pairs; on the Chinese side there are 60,573,223 non-whitespace characters, and the English sentences have 40,629,997 words.

Our main source for training our five-gram language model was the English Gigaword corpus, and we also included close to one million English sentences taken from LDC parallel texts: GALE Year 1 training data (excluding FOUO data), Sinorama, AsiaNet, and Hong Kong news. We restricted the Gigaword corpus to a subsample of 25 million sentences, because of memory constraints.

3 Understanding Chinese Word Segmentation for Phrase-based MT

In this section, we experiment with three types of segmenters – character-based, lexicon-based and feature-based – to explore what kind of characteristics are useful for segmentation for MT.

3.1 Character-based, Lexicon-based and Feature-based Segmenters

The training data for the segmenter is two orders of magnitude smaller than for the MT system, it is not terribly well matched to it in terms of genre and variety, and the information an MT system learns about alignment of Chinese to English might be the basis for a task appropriate segmentation style for Chinese-English MT. A phrase-based MT system like Moses can extract "phrases" (sequences of tokens) from a word alignment and the system can construct the words that are useful. These observations suggest the first hypothesis.

Segmentation Performance			
Segmenter	F measure	OOV Recall	IV Recall
CharBased	0.334	0.012	0.485
MaxMatch	0.828	0.012	0.951
MT Performance			
Segmenter	MT03 (dev)	MT05 (test)	
CharBased	30.81	29.36	
MaxMatch	31.95	30.73	

Table 1: CharBased vs. MaxMatch

Hypothesis 1. *A phrase table should capture word segmentation. Character-based segmentation for MT should not underperform a lexicon-based segmentation, and might outperform it.*

Observation In the experiments we conducted, we found that the phrase table cannot capture everything a Chinese word segmenter can do, and therefore having word segmentation helps phrase-based MT systems. [2]

To show that having word segmentation helps MT, we compare a lexicon-based maximum-matching segmenter with character-based segmentation (treating each Chinese character as a word). The lexicon-based segmenter finds words by greedily matching the longest words in the lexicon in a left-to-right fashion. We will later refer to this segmenter as MaxMatch. The MaxMatch segmenter is a simple and common baseline for the Chinese word segmentation task.

The segmentation performance of MaxMatch is not very satisfying because it cannot generalize to capture words it has never seen before. However, having a basic segmenter like MaxMatch still gives the phrase-based MT system a win over the character-based segmentation (treating each Chinese character as a word). We will refer to the character-based segmentation as CharBased.

In Table 1, we can see that on the Chinese word segmentation task, having MaxMatch is obviously better than not trying to identify Chinese words at all (CharBased). As for MT performance, in Table 1 we see that having a segmenter, even as sim-

[2] Different phrase extraction heuristics might affect the results. In our experiments, grow-diag outperforms both one-to-many and many-to-one for both MaxMatch and CharBased. We report the results only on grow-diag.

ple as MaxMatch, can help phrase-based MT system by about 1.37 BLEU points on all 1082 sentences of the test data (MT05). Also, we tested the performance on 828 sentences of MT05 where all elements are in vocabulary[3] for both MaxMatch and CharBased. MaxMatch achieved 32.09 BLEU and CharBased achieved 30.28 BLEU, which shows that on the sentences where all elements are in vocabulary, there MaxMatch is still significantly better than CharBased. Therefore, Hypothesis 1 is refuted.

Analysis We hypothesized in Hypothesis 1 that the phrase table in a phrase-based MT system should be able to capture the meaning by building "phrases" on top of character sequences. Based on the experimental result in Table 1, we see that using character-based segmentation (CharBased) actually performs reasonably well, which indicates that the phrase table does capture the meaning of character sequences to a certain extent. However, the results also show that there is still some benefit in having word segmentation for MT. We analyzed the decoded output of both systems (CharBased and MaxMatch) on the development set (MT03). We found that the advantage of MaxMatch over CharBased is two-fold, *(i)* lexical: it enhances the ability to disambiguate the case when a character has very different meaning in different contexts, and *(ii)* reordering: it is easier to move one unit around than having to move two consecutive units at the same time. Having words as the basic units helps the reordering model.

For the first advantage, one example is the character "智", which can both mean "intelligence", or an abbreviation for Chile (智利). The comparison between CharBased and MaxMatch is listed in Table 2. The word 失智症 (dementia) is unknown for both segmenters. However, MaxMatch gave a better translation of the character 智. The issue here is not that the "智"→"intelligence" entry never appears in the phrase table of CharBased. The real issue is, when 智 means Chile, it is usually followed by the character 利. So by grouping them together, MaxMatch avoided falsely increasing the probability of translating the stand-alone 智 into Chile. Based on our analysis, this ambiguity occurs the most when the character-based system is dealing with a rare or unseen character sequence in the training data, and also occurs more often when dealing with transliterations. The reason is that characters composing a transliterated foreign named entity usually doesn't preserve their meanings; they are just used to compose a Chinese word that sounds similar to the original word – much more like using a character segmentation of English words. Another example of this kind is "阿耳滋海默氏症" (Alzheimer's disease). The MT system using CharBased segmentation tends to translate some characters individually and drop others; while the system using MaxMatch segmentation is more likely to translate it right.

Reference translation: scientists complete sequencing of the chromosome linked to early dementia
CharBased segmented input: 科␣学␣家␣为␣攸␣关␣初␣期␣失␣智␣症␣的␣染␣色␣体␣完␣成␣定␣序
MaxMatch segmented input: 科学家␣为␣攸关␣初期␣失␣智␣症␣的␣染色␣体␣完成␣定␣序
Translation with CharBased segmentation: scientists at the beginning of the stake of chile lost the genome sequence completed
Translation with MaxMatch segmentation: scientists at stake for the early loss of intellectual syndrome chromosome completed sequencing

Table 2: An example showing that character-based segmentation provides weaker ability to distinguish character with multiple unrelated meanings.

The second advantage of having a segmenter like the lexicon-based MaxMatch is that it helps the reordering model. Results in Table 1 are with the linear distortion limit defaulted to 6. Since words in CharBased are inherently shorter than MaxMatch, having the same distortion limit means CharBased is limited to a smaller context than MaxMatch. To make a fairer comparison, we set the linear distortion limit in Moses to unlimited, removed the lexicalized reordering model, and retested both systems. With this setting, MaxMatch is 0.46 BLEU point better than CharBased (29.62 to 29.16) on MT03. This result suggests that having word segmentation does affect how the reordering model works in a phrase-based system.

Hypothesis 2. *Better Segmentation Performance Should Lead to Better MT Performance*

Observation We have shown in Hypothesis 1 that it is helpful to segment Chinese texts into words first. In order to decide a segmenter to use, the most intuitive thing to do is to find one that gives higher F measure on segmentation. Our experiments show that higher F measure does not necessarily

[3]Except for dates and numbers.

lead to higher BLEU score. In order to contrast with the simple maximum matching lexicon-based model (MaxMatch), we built another segmenter with a CRF model. CRF is a statistical sequence modeling framework introduced by Lafferty et al. (2001), and was first used for the Chinese word segmentation task by Peng et al. (2004), who treated word segmentation as a binary decision task. We optimized the parameters with a quasi-Newton method, and used Gaussian priors to prevent overfitting.

The probability assigned to a label sequence for a particular sequence of characters by a CRF is given by the equation:

$$p_\lambda(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp \sum_{t=1}^{T} \sum_{k=1}^{K} \lambda_k f_k(\mathbf{x}, y_{t-1}, y_t, t) \quad (1)$$

$\mathbf{x}$ is a sequence of T unsegmented characters, $Z(\mathbf{x})$ is the partition function that ensures that Equation 1 is a probability distribution, $\{f_k\}_{k=1}^{K}$ is a set of feature functions, and $\mathbf{y}$ is the sequence of binary predictions for the sentence, where the prediction $y_t = +1$ indicates the t-th character of the sequence is preceded by a space, and where $y_t = -1$ indicates there is none. We trained a CRF model with a set of basic features: character identity features of the current character, previous character and next character, and the conjunction of previous and current characters in the zero-order templates. We will refer to this segmenter as CRF-basic.

Table 3 shows that the feature-based segmenter CRF-basic outperforms the lexicon-based MaxMatch by 5.9% relative F measure. Comparing the OOV recall rate and the IV recall rate, the reason is that CRF-basic wins a lot on the OOV recall rate. We see that a feature-based segmenter like CRF-basic clearly has stronger ability to recognize unseen words. On MT performance, however, CRF-basic is 0.38 BLEU points worse than MaxMatch on the test set. In Section 3.2, we will look at how the MT training and test data are segmented by each segmenter, and provide statistics and analysis for why certain segmenters are better than others.

3.2 Consistency Analysis of Different Segmenters

In Section 3.1 we have refuted two hypotheses. Now we know that: (*i*) phrase table construction does not fully capture what a word segmenter can do. Thus it is useful to have word segmentation for MT. (*ii*) a higher F measure segmenter does not necessarily outperforms on the MT task.

Segmentation Performance

Segmenter	F measure	OOV Recall	IV Recall
CRF-basic	0.877	0.502	0.926
MaxMatch	0.828	0.012	0.951
CRF-Lex	0.940	0.729	0.970

MT Performance

Segmenter	MT03 (dev)	MT05 (test)
CRF-basic	33.01	30.35
MaxMatch	31.95	30.73
CRF-Lex	32.70	30.95

Table 3: CRF-basic vs MaxMatch

Segmenter	#MT Training Lexicon Size	#MT Test Lexicon Size
CRF-basic	583147	5443
MaxMatch	39040	5083
CRF-Lex	411406	5164
	MT Test Lexicon OOV rate	Conditional Entropy
CRF-basic	7.40%	0.2306
MaxMatch	0.49%	0.1788
CRF-Lex	4.88%	0.1010

Table 4: MT Lexicon Statistics and Conditional Entropy of Segmentation Variations of three segmetners

To understand what factors other than segmentation F measure can affect MT performance, we introduce another CRF segmenter CRF-Lex that includes lexicon-based features by using external lexicons. More details of CRF-Lex will be described in Section 5.1. From Table 3, we see that the segmentation F measure is that CRF-Lex > CRF-basic > MaxMatch. And now we know that the better segmentation F measure does not always lead to better MT BLEU score, because of in terms of MT performance, CRF-Lex > MaxMatch > CRF-basic.

In Table 4, we list some statistics of each segmenter to explain this phenomenon. First we look at the lexicon size of the MT training and test data. While segmenting the MT data, CRF-basic generates an MT training lexicon size of 583K unique word tokens, and MaxMatch has a much smaller lexicon size of 39K. CRF-Lex performs best on MT, but the MT training lexicon size and test lexicon OOV rate is still pretty high compared to MaxMatch. Only examining the MT training and test lexicon size still doesn't fully explain why CRF-Lex outperforms MaxMatch. MaxMatch generates a smaller MT lexicon and lower OOV rate, but for MT it wasn't better than CRF-Lex, which has a bigger lexicon and higher OOV rate. In order to understand why MaxMatch performs worse on MT than CRF-Lex but better than CRF-basic, we use conditional entropy of segmentation variations to measure consistency.

We use the gold segmentation of the SIGHAN test data as a guideline. For every work type w_i, we collect all the different pattern variations v_{ij} in the segmentation we want to examine. For example, for a word "*ABC*" in the gold segmentation, we look at how it is segmented with a segmenter. There are many possibilities. If we use c_x and c_y to indicate other Chinese characters and ␣ to indicate white spaces, "c_x␣ABC␣c_y" is the correct segmentation, because the three characters are properly segmented from both sides, and they are concatenated with each other. It can also be segmented as "c_x␣A␣BC␣c_y", which means although the boundary is correct, the first character is separated from the other two. Or, it can be segmented as "c_xA␣BCc_y", which means the first character was actually part of the previous word, while *BC* are the beginning of the next word. Every time a particular word type w_i appears in the text, we consider a segmenter more consistent if it can segment w_i in the same way every time, but it doesn't necessarily have to be the same as the gold standard segmentation. For example, if "*ABC*" is a Chinese person name which appears 100 times in the gold standard data, and one segmenter segment it as c_x␣A␣BC␣c_y 100 times, then this segmenter is still considered to be very consistent, even if it doesn't exactly match the gold standard segmentation. Using this intuition, the conditional entropy of segmentation variations $H(V|W)$ is defined as follows:

$$\begin{aligned} H(V|W) &= -\sum_{w_i} P(w_i) \sum_{v_{ij}} P(v_{ij}|w_i) \log P(v_{ij}|w_i) \\ &= -\sum_{w_i} \sum_{v_{ij}} P(v_{ij}, w_i) \log P(v_{ij}|w_i) \end{aligned}$$

Now we can look at the overall conditional entropy $H(V|W)$ to compare the consistency of each segmenter. In Table 4, we can see that even though MaxMatch has a much smaller MT lexicon size than CRF-Lex, when we examine the consistency of how MaxMatch segments in context, we find the conditional entropy is much higher than CRF-Lex. We can also see that CRF-basic has a higher conditional entropy than the other two. The conditional entropy $H(V|W)$ shows how consistent each segmenter is, and it correlates with the MT performance in Table 4. Note that consistency is only one of the competing factors of how good a segmentation is for MT performance. For example, a character-based segmentation will always have the best consistency possible, since every word *ABC* will just have one pattern: c_x␣A␣B␣C␣c_y. But from Section 3.1 we see that CharBased performs worse than both MaxMatch and CRF-basic on MT, because having word segmentation can help the granularity of the Chinese lexicon match that of the English lexicon.

In conclusion, for MT performance, it is helpful to have consistent segmentation, while still having a word segmentation matching the granularity of the segmented Chinese lexicon and the English lexicon.

4 Optimal Average Token Length for MT

We have shown earlier that word-level segmentation vastly outperforms character based segmentation in MT evaluations. Since the word segmentation standard under consideration (Chinese Treebank (Xue et al., 2005)) was neither specifically designed nor optimized for MT, it seems reasonable to investigate whether any segmentation granularity in continuum between character-level and CTB-style segmentation is more effective for MT. In this section, we present a technique for directly optimizing a segmentation property—characters per token average—for translation quality, which yields significant improvements in MT performance.

In order to calibrate the average word length produced by our CRF segmenter—i.e., to adjust the rate of word boundary predictions ($y_t = +1$), we apply a relatively simple technique (Minkov et al., 2006) originally devised for adjusting the precision/recall tradeoff of any sequential classifier. Specifically, the weight vector $\mathbf{w}$ and feature vector of a trained linear sequence classifier are augmented at test time to include new class-conditional feature functions to bias the classifier towards particular class labels. In our case, since we wish to increase the frequency of word boundaries, we add a feature function:

$$f_0(\mathbf{x}, y_{t-1}, y_t, t) = \begin{cases} 1 & \text{if } y_t = +1 \\ 0 & \text{otherwise} \end{cases}$$

Its weight λ_0 controls the extent of which the classifier will make positive predictions, with very large positive λ_0 values causing only positive predictions (i.e., character-based segmentation) and large negative values effectively disabling segmentation boundaries. Table 5 displays how changes of the

λ_0	-1	0	1	2	4	8	32
len	1.64	1.62	1.61	1.59	1.55	1.37	1

Table 5: Effect of the bias parameter λ_0 on the average number of character per token on MT data.

bias parameter λ_0 affect segmentation granularity.[4] Since we are interested in analyzing the different regimes of MT performance between CTB segmentation and character-based, we performed a grid search in the range between $\lambda_0 = 0$ (maximum-likelihood estimate) and $\lambda_0 = 32$ (a value that is large enough to produce only positive predictions). For each λ_0 value, we ran an entire MT training and testing cycle, i.e., we re-segmented the entire training data, ran GIZA++, acquired phrasal translations that abide to this new segmentation, and ran MERT and evaluations on segmented data using the same λ_0 values.

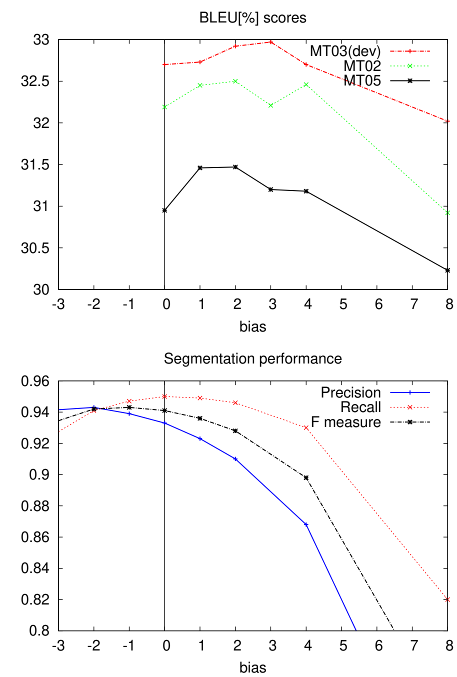

Figure 1: A bias towards more segment boundaries ($\lambda_0 > 0$) yields better MT performance and worse segmentation results.

Segmentation and MT results are displayed in Figure 1. First, we observe that an adjustment of the precision and recall tradeoff by setting negative bias values ($\lambda_0 = -2$) slightly improves segmentation performance. We also notice that raising λ_0 yields relatively consistent improvements in MT performance, yet causes segmentation performance (F measure) to be increasingly worse. While the latter finding is not particularly surprising, it further confirms that segmentation and MT evaluations can yield rather different outcomes. We chose the $\lambda_0 = 2$ on another dev set (MT02). On the test set MT05, $\lambda_0 = 2$ yields 31.47 BLEU, which represents a quite large improvement compared to the unbiased segmenter (30.95 BLEU). Further reducing the average number of characters per token yields gradual drops of performance until character-level segmentation ($\lambda_0 \geq 32$, 29.36 BLEU).

Here are some examples of how setting $\lambda_0 = 2$ shortens the words in a way that can help MT.

- separating adjectives and pre-modifying adverbs: 很大(*very big*) → 很(*very*) 大(*big*)
- separating nouns and pre-modifying adjectives: 高血压(*high blood pressure*) → 高(*high*) 血压(*blood pressure*)
- separating compound nouns: 内政部(*Department of Internal Affairs*) → 内政(*Internal Affairs*) 部(*Department*).

5 Improving Segmentation Consistency of a Feature-based Sequence Model for Segmentation

In Section 3.1 we showed that a statistical sequence model with rich features can generalize better than maximum matching segmenters. However, it also inconsistently over-generates a big MT training lexicon and OOV words in MT test data, and thus causes a problem for MT. To improve a feature-based sequence model for MT, we propose 4 different approaches to deal with named entities, optimal length of word for MT and joint search for segmentation and MT decoding.

5.1 Making Use of External Lexicons

One way to improve the consistency of the CRF model is to make use of external lexicons (which are not part of the segmentation training data) to add lexicon-based features. All the features we use are listed in Table 6. Our linguistic features are adopted from (Ng and Low, 2004) and (Tseng et al., 2005). There are three categories of features:

[4]Note that character-per-token averages provided in the table consider each non-Chinese word (e.g., foreign names, numbers) as one character, since our segmentation post-processing prevents these tokens from being segmented.

Lexicon-based Features	Linguistic Features
(1.1) $L_{Begin}(C_n), n \in [-2,1]$	(2.1) $C_n, n \in [-2,1]$
(1.2) $L_{Mid}(C_n), n \in [-2,1]$	(2.2) $C_{n-1}C_n, n \in [-1,1]$
(1.3) $L_{End}(C_n), n \in [-2,1]$	(2.3) $C_{n-2}C_n, n \in [1,2]$
(1.4) $L_{End}(C_{-1}) + L_{End}(C_0) + L_{End}(C_1)$	(2.4) $Single(C_n), n \in [-2,1]$
(1.5) $L_{End}(C_{-2}) + L_{End}(C_{-1}) + L_{Begin}(C_0) + L_{Mid}(C_0)$	(2.5) $UnknownBigram(C_{-1}C_0)$
(1.6) $L_{End}(C_{-2}) + L_{End}(C_{-1}) + L_{Begin}(C_{-1}) + L_{Begin}(C_0) + L_{Mid}(C_0)$	(2.6) $ProductiveAffixes(C_{-1}, C_0)$
	(2.7) $Reduplication(C_{-1}, C_n), n \in [0,1]$

Table 6: Features for CRF-Lex

Segmentation Performance			
Segmenter	F measure	OOV Recall	IV Recall
CRF-Lex-NR	0.943	0.753	0.970
CRF-Lex	0.940	0.729	0.970
MT Performance			
Segmenter	MT03 (dev)	MT05 (test)	
CRF-Lex-NR	32.96	31.27	
CRF-Lex	32.70	30.95	

Table 7: CRF-Lex-NR vs CRF-Lex

character identity n-grams, morphological and character reduplication features. Our lexicon-based features are adopted from (Shi and Wang, 2007), where $L_{Begin}(C_0)$, $L_{Mid}(C_0)$ and $L_{End}(C_0)$ represent the maximum length of words found in a lexicon that contain the current character as either the first, middle or last character, and we group any length equal or longer than 6 together. The linguistic features help capturing words that were unseen to the segmenter; while the lexicon-based features constrain the segmenter with external knowledge of what sequences are likely to be words.

We built a CRF segmenter with all the features listed in Table 6 (CRF-Lex). The external lexicons we used for the lexicon-based features come from various sources including named entities collected from Wikipedia and the Chinese section of the UN website, named entities collected by Harbin Institute of Technology, the ADSO dictionary, EMM News Explorer, Online Chinese Tools, Online Dictionary from Peking University and HowNet. There are 423,224 distinct entries in all the external lexicons.

The MT lexicon consistency of CRF-Lex in Table 4 shows that the MT training lexicon size has been reduced by 29.5% and the MT test data OOV rate is reduced by 34.1%.

5.2 Joint training of Word Segmentation and Proper Noun Tagging

Named entities are an important source for OOV words, and in particular are ones which it is bad to break into pieces (particularly for foreign names). Therefore, we use the proper noun (NR) part-of-speech tag information from CTB to extend the label sets of our CRF model from 2 to 4 ({beginning of a word, continuation of a word} × {NR, not NR}). This is similar to the "all-at-once, character-based" POS tagging in (Ng and Low, 2004), except that we are only tagging proper nouns. We call the 4-label extension CRF-Lex-NR. The segmentation and MT performance of CRF-Lex-NR is listed in Table 7. With the 4-label extension, the OOV recall rate improved by 3.29%; while the IV recall rate stays the same. Similar to (Ng and Low, 2004), we found the overall F measure only goes up a tiny bit, but we do find a significant OOV recall rate improvement.

On the MT performance, CRF-Lex-NR has a 0.32 BLEU gain on the test set MT05. In addition to the BLEU improvement, CRF-Lex-NR also provides extra information about proper nouns, which can be combined with postprocessing named entity translation modules to further improve MT performance.

6 Conclusion

In this paper, we investigated what segmentation properties can improve machine translation performance. First, we found that neither character-based nor a standard word segmentation standard are optimal for MT, and show that an intermediate granularity is much more effective. Using an already competitive CRF segmentation model, we directly optimize segmentation granularity for translation quality, and obtain an improvement of 0.73 BLEU point on MT05 over our lexicon-based segmentation baseline. Second, we augment our CRF model with lexicon and proper noun features in order to improve segmentation consistency, which provide a 0.32 BLEU point improvement.

7 Acknowledgement

The authors would like to thank Menqgiu Wang and Huihsin Tseng for useful discussions. This paper is based on work funded in part by the Defense Advanced Research Projects Agency through IBM.

References

Jianfeng Gao, Mu Li, Andi Wu, and Chang-Ning Huang. 2005. Chinese word segmentation and named entity recognition: A pragmatic approach. *Computational Linguistics*.

Nizar Habash and Fatiha Sadat. 2006. Arabic preprocessing schemes for statistical machine translation. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pages 49–52, New York City, USA, June. Association for Computational Linguistics.

Vera Hollink, Jaap Kamps, Christof Monz, and Maarten de Rijke. 2004. Monolingual document retrieval for European languages. *Information Retrieval*, 7(1).

Philipp Koehn and Kevin Knight. 2003. Empirical methods for compound splitting. In *EACL '03: Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics*, pages 187–193. Association for Computational Linguistics.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proc. of NAACL-HLT*.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. 18th International Conf. on Machine Learning*.

Gina-Anne Levow. 2006. The third international Chinese language processing bakeoff: Word segmentation and named entity recognition. In *Proc. of the Fifth SIGHAN Workshop on Chinese Language Processing*, July.

Einat Minkov, Richard Wang, Anthony Tomasic, and William Cohen. 2006. NER systems that suit user's preferences: Adjusting the recall-precision trade-off for entity extraction. In *Proc. of NAACL-HLT, Companion Volume: Short Papers*, New York City, USA, June.

Hwee Tou Ng and Jin Kiat Low. 2004. Chinese part-of-speech tagging: One-at-a-time or all-at-once? Word-based or character-based? In *Proc. of EMNLP*.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1).

Franz Josef Och. 2003. Minimum error rate training for statistical machine translation. In *ACL*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2001. BLEU: a method for automatic evaluation of machine translation. In *ACL*.

Fuchun Peng, Xiangji Huang, Dale Schuurmans, and Nick Cercone. 2002. Investigating the relationship between word segmentation performance and retrieval performance in Chinese IR. In *Proc. of the 19th International Conference on Computational Linguistics*.

Fuchun Peng, Fangfang Feng, and Andrew McCallum. 2004. Chinese segmentation and new word detection using conditional random fields. In *Proc. of COLING*.

Yanxin Shi and Mengqiu Wang. 2007. A dual-layer CRFs based joint decoding method for cascaded segmentation and labeling tasks. In *IJCAI*.

Huihsin Tseng, Pichuan Chang, Galen Andrew, Daniel Jurafsky, and Christopher Manning. 2005. A conditional random field word segmenter for Sighan bakeoff 2005. In *Proc. of the Fourth SIGHAN Workshop on Chinese Language Processing*.

Jia Xu, Richard Zens, and Hermann Ney. 2004. Do we need Chinese word segmentation for statistical machine translation. In *Proc. of the Third SIGHAN Workshop on Chinese Language Learning*.

Nianwen Xue, Fei Xia, Fu dong Chiou, and Martha Palmer. 2005. Building a large annotated Chinese corpus: the Penn Chinese treebank. *Journal of Natural Language Engineering*, 11(2).