A Very Very Preliminary Investigation of the Effects of Segmentation on Chinese Phrased-Based Statistical Machine Translation Performance

William Morgan

Abstract

I investigate the effect of different Chinese segmentation standards on final machine translation performance. I train a statistical phrase-based Chinese to English machine translation model, using GIZA++ for word alignment, SRILM for language model generation, and Philipp Köhn's "training" package for phrase extraction. I segment the Chinese text according to four different methods: LDC, PKU, simple per-character, and cross-lingual word alignment (using the output of the per-character word alignment to segment). I evaluate system performance on training corpora of different sizes. I calculate BLEU score against the NIST 2003 training set and compare our system performance against 2003 state-of-the-art. I show, somewhat surprisingly, segmenting according to the LDC standard yields the highest performance.

1 Introduction

Statistical machine translation (SMT) is a technique for automatically translating between (typically resource-rich) language pairs, focused on extracting information necessary for translation directly from data. Since its origins in the 1990's (Brown et al., 1990), the basic approach has been to make use of parallel corpora (bilingual corpora with (ideally) a one-to-one correspondence between sentences) by aligning words between corresponding sentence pairs.

Phrase-based SMT (Och, 2002; Koehn et al., 2003) have recently emerged as a very successful technique that operates on the phrase, rather than word, level.[1] Typically such systems rely on standard SMT algorithms to produce an initial word alignment, and build up phrases based on that. Phrase-based SMT performance has placed at the top of recent NIST evaluations.

Phrase-based SMT thus makes heavy use of the notion of a word. In order to train such a system for translating to or from languages that do not naturally segment sentences into words, such as Chinese or Japanese, it is then necessary to segment the input.

In some unsegmented languages, such as Japanese, the notion of a word is never the less linguistically well-defined, and the prevailing approach in SMT is to segment according to linguistic principles. However, in a language such as Chinese, the question of whether a sequence of characters should be considered a word is word is not always clear, and consequently many different segmentations are plausible for a given sentence.

In the absence, then, of firm linguistic principles, it is natural to ask: what segmentation is *best* for the task? Recent work in Chinese named-entity recognition has also begun to explore this question. (Gao et al., 2006).

The best segmentation for a machine translation system is the segmentation that, *ceteris paribus*, yields the highest quality output. I address this question by evaluating the final performance of a phrase-based SMT system when trained and evaluated on a Chinese-English translation task with four different methods of segmentation.

2 Segmentation

In this study I focus on four different types of segmentation: one baseline, two linguistically motivated, and one based on the parallel text. Example segmentations can be seen in figure 2.

Character: a baseline segmentation scheme in which every character is considered its own word.

LDC: the segmentation standard used by the Linguistic Data Consortium (LDC) at the University of Pennsylvania. (Xia, 2000) I use a conditional random field (CRF) segmenter from Tseng et al. (2005), trained on the Chinese Treebank, (Xue et al., 2002) to segment under this scheme.

[1] Where "phrase" here means simply a sequence of words, i.e. carries no linguistic connotations.

The LDC segmentation style is typically fairly agglomerative; things like person names and nationalities are typically kept together.

PKU: the segmentation standard used by Peking University. I use the CRF segmenter from Tseng et al., trained on the SIGHAN 2005 bakeoff data, to segment under this scheme.

PKU segmentation splits more frequently than LDC; for example, person names have the surname segmented (as in English) and nationalities have the final "person" character segmented off (*not* as in English).

"Aligned": segmentation based on word alignment between Chinese characters and English words. This is performed in the following manner.

- An alignment is created between Chinese characters to English words using GIZA++ (see section 4), such that each English word is aligned to zero or more Chinese characters, and each Chinese character is aligned to exactly one English word, including the "NULL" word (indicating no alignment).
- All consecutive Chinese characters that are aligned to the same English word are joined together to form a pseudo-word. (I.e. one English word can "produce" multiple Chinese pseudo-words if it is aligned with non-consecutive characters.) All characters aligned to NULL are inserted as individual psuedo-words.
- The pseudo-words are sorted so as to appear in the same order as the characters in the original sentence.

Note that this sequence of operations guarantees that the resulting character sequence is the same as in the original Chinese sentence, i.e. the algorithm produces a segmentation. Pseudo-code for this algorithm is in figure 1.

The segmentation produced by this process is quite different from the LDC and PKU segmentation, as one might expect. When the alignment algorithm is off, the segmentation is tragically incorrect; when the alignment algorithm works well, the segmentation makes sense, but would strike Chinese readers as highly unusual. For example, the two characters for Hong Kong are typically broken apart, whereas those for Beijing are kept together.

3 Data

I use the Hong Kong parallel text corpus (LDC publication LDC2004T08) to train the SMT system. This corpus has three components: hansards, laws and news. For the purposes of this study I use only the news corpus, under the assumption that it is the least specialized. This corpus contains roughly 700,000 sentence pairs, consisting of 17m English words and 23m Chinese characters.

All Chinese data was converted from traditional characters in the `BIG5-HKSCS` encoding to simplified characters in `UTF-8` encoding, as the evaluation data set is in `UTF-8`.

In the interest of comparing the effect of training corpus size of system performance, I experiment with different subsets of the news corpus. Table 1 describes the subsets.

The English side of the data was downcased and the English side underwent minimal tokenization in preparation for word alignment.

initialize words $w \leftarrow \emptyset$
for all aligned English words e_i **do**
  let c be the set of Chinese chars aligned with e
  for all consecutive sequences s_j in c **do**
    $w \leftarrow w + s_j$
  end for
end for
for all characters c_i aligned to NULL **do**
  $w \leftarrow w + s_j$
end for
sort w by character index of first character
return w

Figure 1: Pseudo-code for generating a segmentation from a word alignment.

4 Experimental Setup

A phrase-based SMT system consists of several parts. Happily, each of these is readily available for research purposes.

Word alignment: I use GIZA++[4] (Och and Ney, 2003), an implementaiton of the IBM SMT models, for word alignment.[5] I run GIZA++ to learn IBM model 4 translation tables in both directions, English to Chinese and Chinese to English, and the word alignments produced are then extracted to build a phrase table.

The GIZA++ Chinese to English word alignment with character segmentation is also used to generate the "aligned" segmentation (see section 2 for details).

[4] Now installed in `/u/nlp/packages/GIZA++/`.

[5] GIZA++ has not been updated for a few years, so non-trivial effort was required to get it (and associated tools like mkcls) to compile with a modern C++ compiler. I submitted a patch to Franz Och, but he did not reply.

LDC: 一 _ 项 _ 庄严 _ 肃穆 _ 富 _ 历史 _ 意义 _ 的 _ 仪式 _ 标志 _ 中华 _ 人民 _ 共和国 _ 对 _ 香港 _ 恢复 _ 行使 _ 主权 _ 。

PKU: 一 _ 项 _ 庄 _ 严肃 _ 穆富 _ 历史 _ 意义 _ 的 _ 仪式 _ 标志 _ 中华人民共和国 _ 对 _ 香港 _ 恢复 _ 行使 _ 主权 _ 。

aligned: 一 _ 项庄严肃穆富 _ 历史意义 _ 的 _ 仪式 _ 标志 _ 中华人民共和国 _ 对 _ 香 _ 港 _ 恢复 _ 行使主权 _ 。

English: a solemn , historic ceremony has marked the resumption of the exercise of sovereignty over hong kong by the people 's republic of china .

Figure 2: Example segmentations for a Chinese sentence. The LDC and PKU segmentations differ how they treat “The People’s Repulic of China” (PKU segmenting off “China”, LDC keeping it attached). The aligned segmentation also keeps it attached, but splits the two characters for “Hong Kong” (presumably because each aligns with an English word).[3]

corpus	sentence pairs	English	LDC	PKU	chars	aligned
full	698,688	17,124,632	14,893,502	14,705,143	23,644,293	
300k	300,000	7,242,993	6,304,757	6,215,329	10,212,544	
70k	70,000	1,746,697	1,517,552	1,496,566	2,449,443	1,368,434
7k	7,000	186,941	160,751	158,698	260,888	144,212

Table 1: Number of sentence pairs and words for the four corpora subsets tried. (Some values missing due to time constraints.) “English” refers to the number of words on the English side; “LDC”, “PKU”, “chars” and “aligned” refer to the number of words on the Chinese side under that tokenization scheme. (Thus “chars” is conveniently also the number of characters on the Chinese side.)

Phrase tables: Given alignments in both directions, I use Philipp Köhn's "training" package to build phrase tables in a heuristic manner. The package provides a variety of heuristics; I use the "`GROW-DIAG-FINAL`" method. This method starts with the intersection of the two alignments and successively adds points from the union that are neighbors of added points. As a final ("`-FINAL`") step, it adds any unaligned words that are in the union.

No attempt was made to explore any of the other phrase table heuristics.

The package also does the "dirty work" of determining lexical translation probabilities, scoring the phrases based on these and phrase translation probabilities, etc.

Language model: I use the SRILM toolkit[6] to produce a trigram language model with modified Kneser-Ney smoothing from the English side of the (full) corpus.

Decoding: I used Pharaoh[7] (Köhn, 2004) as the decoder, leaving all parameters set at their default.

Model tuning : I compiled Carmel[8] (a finite-state toolkit) and experimented with the minimum-error-rate training provided in Köhn's package, but without success. Thus, the models are used with the default decoding parameters.

5 Evaluation

The complete system was trained on each of the four corpus sizes under each of the four segmentation types.

For evaluation, I use the BLEU metric, as implemented in the Köhn training package, and the NIST 2003 Machine Translation Workshop evaluation corpus (LDC publication LDC2006T04). This corpus consists of 919 Chinese newswire sentences, in simplified characters, and four reference translations for each.

The evaluation corpus also contains the system output of several systems submitted to the workshop, allowing comparisons to be made against 2003 state-of-the-art technology.

Since the translation model developed in this work contains no true-casing, I downcased all English data (both reference and system output) in this corpus in order to perform a case-insensitive evaluation.

The evaluation procedure is as follows. For each segmentation type, the evaluation corpus text is segmented according to that standard, the system is run on that input, and the output is then compared to the four reference translations using BLEU.

[6]Now installed in `/u/nlp/packages/SRILM/`.

[7]Now installed in `/u/nlp/packages/pharaoh/`.

[8]Now installed in `/u/nlp/packages/carmel/`.

	LDC	PKU	chars	aligned
full	10.67	10.58		
300k	9.85	9.72		
70k	8.67	8.35	7.27	7.29
7k	5.14	5.24	4.20	4.75

Table 2: BLEU score per corpus per segmentation type. (Some values missing due to time constraints.)

One issue remains with the "aligned" segmentation type, in that we must ideally segment the evaluation data in a similar manner before we can begin decoding. There are a couple ways one might do this:

- Align the evaluation data with GIZA++ and segment using the algorithm in figure 1.
- Train the CRF segmenter on the "aligned"-segmented version of the entire news corpus, and use this to segment the evaluation data.
- Simply use another segmentation standard and hope that it works out.

Option 5 is perhaps not in the spirit of the game, as real-life input could not be so segmented. Option 5 is thus probably the best approach. What I did, of course, was option 5, in the interest of saving time. Using the 7k corpus, I tried LDC, PKU and character segmentation of the evaluation corpus Chinese side, fed this to the aligned-segmented model, and determined that LDC segmentation produced the highest BLEU score. Thus the numbers for this model in section 6 may be artificially low.

6 Results

Unfortunately, despite the ready availability of these packages, getting them all to compile and work together took a significant amount of effort. Furthermore, the character-segmented corpus in particular required an inordinate amount of time to train—44 hours on the full corpus.[9] GIZA++ is the main time sink, especially with the large number of "words" in the character-segmented corpus.

That said, table 2 gives the results of the experiments that were completed in time. Figure 3 is a graphical version of the same data.

For purposes of comparison, other system output from the NIST 2003 evaluation ranges from 4.96 to 24.66, with 4 of the 9 systems under our high score of 10.67. It

[9]And then an error was discovered!

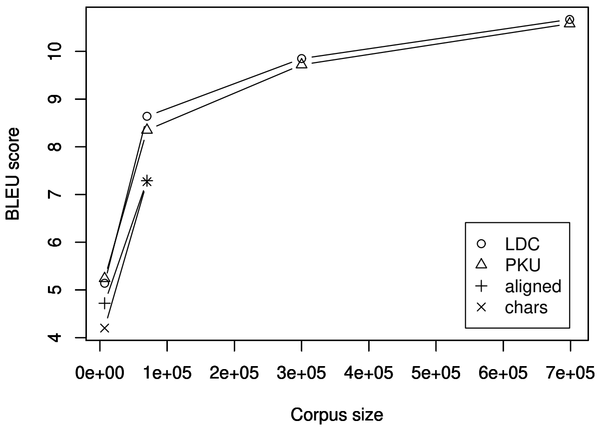

Figure 3: BLEU score per corpus per segmentation type. (Some values missing due to time constraints.)

is likely that system performance would be further improved by model tuning and by an even larger training corpus.

7 Analysis

It is clear from table 2 that LDC segmentation gives the best final performance, although PKU segmentation is not too far behind, and, in the case of the smallest corpus, actually surpasses LDC performance.

Upon examination of the phrase tables, it becomes clear that the character model captures much smaller phrases than the word-based models. Furthermore, it captures far *fewer* phrases—for the 70k corpus, for example, while LDC and PKU segmentation capture roughly 630k phrases, and the aligned segmentation 660k phrases, the character-based segmentation phrase table contains only 540k entries. This may be part of the reason that character-based performance is so low.

The difference between LDC and PKU segmentation is somewhat surprising, as the intution is that PKU segmentation, being more "verbose", should be better for MT. But an examination of table 1 shows that, the two examples in section 2 aside, LDC segmentation actually tends to be slightly more verbose that PKU segmentation. This may play a factor in the difference between the two, although it's not that clear (to me!) that verbosity is actually beneficial for a phrase-based system.

The poor performance of the aligned segmentation is disappointing. In particular, since the phrase tables generated are much larger relative to the phrase tables in the other systems, it is surprising that performance is so low. One possible explanation is that the phrase table, while large, is quite poor—many of the character alignments produced by GIZA++ for the character segmentation model are incorrect, and, while clumping characters together and re-running GIZA++ does improve performance somewhat, the phrase table may still have many errors. I suspect it is more likely that the use of LDC segmentation has a significantly detrimental affect, especially since the two segmentation standards are radically different—it is likely that, despite its size, vast portions of the table are unusable as they simply correspond to "words" that are never generated by the LDC segmentater.

8 Future work

More analysis of the above results is required, particularly on the aligned segmentation model. A segmenter should be trained on this output and used to segment the evaluation corpus.

It is also possible to play with model tuning and different phrase heuristics, and likely that performance can be improved significantly by doing so. For exaple, phrase heuristics may turn out have particular behaviors when aligning words and characters.

There is a significant genre gap between the training corpus and the evaluation corpus here. Addressing this issue may also improve performance.

Training the language model on more English data is immediately doable and would likely improve performance.

Acknowledgments

The author would like to thank Pi-Chuan Chang for helpful discussion of the differences between segmentation styles, and for assistance in running the CRF segmenter; and Chris Manning, for providing the idea behind the alignment-based segmentation.

References

Peter F. Brown, John Cocke, Stephen A. Della Pietra, Vincent J. Della Pietra, Fredrick Jelinek, John D. Lafferty, Robert L. Mercer, and Paul S. Roossin. 1990. A statistical approach to machine translation. *Computational Linguistics*, 16(2):79–85.

Jianfeng Gao, Mu Li, Andi Wu, and Chang-Ning Huang. 2006. Chinese word segmentation and named entity recognition: A pragmatic approach. *Computational Linguistics*, 31(4):531–574.

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

Philipp Köhn. 2004. Pharaoh: A beam search decoder for phrase-based statistical machine translation models. In *Proceedings of the AMTA*.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Franz J. Och. 2002. *Statistical Machine Translation: From Single=word models to alignment tempaltes*. Ph.D. thesis, RWTH Aachen.

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

Fei Xia. 2000. The segmentation guidelines for the Penn Chinese Treebank (3.0). technical report 00-06, Institute for Research in Cognitive Science, University of Pennsylvania.

Nianwen Xue, Fu-Dong Chiou, and Martha Palmer. 2002. Building a large-scale annotated chinese corpus. In *Proceedings of the 19th International Conference on Computational Linguistics*.