Pre-processing of Bilingual Corpora for Mandarin-English EBMT

Ying Zhang, Ralf Brown, Robert Frederking, Alon Lavie

Language Technologies Institute, Carnegie Mellon University
NSH, 5000 Forbes Ave. Pittsburgh, PA 15213
U.S.A
{joy+, ralf+, ref+,alavie+}@cs.cmu.edu

Abstract

Pre-processing of bilingual corpora plays an important role in Example-Based Machine Translation (EBMT) and Statistical-Based Machine Translation (SBMT). For our Mandarin-English EBMT system, pre-processing includes segmentation for Mandarin, bracketing for English and building a statistical dictionary from the corpora. We used the Mandarin segmenter from the Linguistic Data Consortium (LDC). It uses dynamic programming with a frequency dictionary to segment the text. Although the frequency dictionary is large, it does not completely cover the corpora. In this paper, we describe the work we have done to improve the segmentation for Mandarin and the bracketing process for English to increase the length of English phrases. A statistical dictionary is built from the aligned bilingual corpus. It is used as feedback to segmentation and bracketing to re-segment / re-bracket the corpus. The process iterates several times to achieve better results. The final results of the corpus pre-processing are a segmented/bracketed aligned bilingual corpus and a statistical dictionary. We achieved positive results by increasing the average length of Chinese terms about 60% and 10% for English. The statistical dictionary gained about a 30% increase in coverage.

Keywords

EBMT, Bilingual corpus, pre-processing, tokenization, bilingual segmentation/bracketing

1 Background

We describe the pre-processing of Mandarin-English bilingual corpora for an Example-Based Machine Translation (EBMT) system.

The EBMT software (Brown, 1996; Brown 1999) used for the experiments described here is a shallow system which can function using nothing more than sentence-aligned plain text and a bilingual dictionary; and given sufficient parallel text, the dictionary can be extracted statistically from the corpus. Details will be given in section 6. To perform a translation, the program looks up all matching phrases in the source language half of the parallel corpus and performs a word-level alignment on the entries containing matches to determine a (usually partial) translation. Portions of the input for which there are no matches in the corpus do not generate a translation. Because the EBMT system does not generate translations for 100% of the input text, a bilingual dictionary and phrasal glossary are used to fill the gaps. Selection of a "best" translation is guided by a trigram model of the target language (Hogan & Frederking, 1998).

As our EBMT (as well as dictionary and glossary) approaches are word-based, but Chinese is ordinarily written without spaces between words, Chinese input must be segmented into individual words. In the initial baseline system, the segmenter used for corpus pre-processing is provided by the Linguistic Data Consortium (LDC). The LDC segmenter tries to find the segmentation with the fewest words, guided by the highest product of word frequency. Provided with the segmenter is a word frequency list of Mandarin words. Although this list is large, it does not completely cover the vocabulary of the EBMT corpora (described below). As a result, many sentences had mis-segmentation errors where a Chinese word was not recognized and broken into single Chinese characters or small chunks. Among all kinds of segmentation errors, mis-segmentation (about 1.43% per token) is much more frequent than incorrect segmentation (about 0.142% per token), where the segmenter makes wrong decisions at ambiguous word boundaries.

2 Data and Definitions

Data

The bilingual corpus we used for the experiments is the Hong Kong News Parallel corpus (HKnews), provided by the LDC. After cleaning, 90% of the data is used for training, 5% for the development test set (dev-test) and 5% for the test set. Table 1 shows some features of this corpus. The "Word Type" and "Word Token" are calculated after segmenting the Chinese part using the LDC segmenter.

	Training	Dev-test	Test
Size	24.53 MB	1.27 MB	1.27 MB
Sent.	95,752	4,992	4,866
Chinese Word Type	20,451	8,529	8,511
Chinese Word Token	2,600,095	134,749	135,372

Table 1: Corpus Features

Definitions

Here we give the definitions of the terminology used in this paper:

Chinese Characters
The smallest unit in written Chinese is a character, which is represented by 2 bytes in GB-2312 code.

Chinese Words
A word in natural language is the smallest reusable unit which can be used in isolation. A Chinese word can be

one character or a sequence of characters. The definition of Chinese words is vague, especially when one tries to decide whether a sequence is a word or a phrase.

A word in Chinese is usually a bigram (two characters), but may also be a unigram, a trigram, or a four-gram. Function words are often unigrams, and n-grams with n>4 are usually specific idioms. According to the Frequency Dictionary of Modern Chinese (FDMC 1986), among the top 9000 most frequent words, 26.7% are unigrams, 69.8% are bigrams, 2.7% are trigrams, 0.007% four-grams, and 0.0002% 5-grams.

Chinese Phrases
We define a Chinese phrase as a sequence of Chinese words. For each word in the phrase, the meaning of this word is the same as the meaning when the word appears by itself. In a sequence of characters $S=C_1\ C_2\ C_3\ \ldots C_i\ C_{i+1}\ldots\ C_n$, for a particular segmentation Θ, which segments the sequence into p words: $W_1\ (start_1=1,\ end_1)$, $W_2(start_2=end_1+1,\ end_2)\ ,\ldots,\ W_i(start_i=end_{i-1}+1,\ end_{i+1})\ ,\ldots,\ W_p(start_p=end_{p-1}+1,\ end_p=n)$, if for every word W_i $(1 \le i \le p)$, its meaning in sequence S is the same as the meaning when it appears in other contexts, then S is a phrase.

Terms
A term is a meaningful constituent. It can be either a word or a phrase.

Segmentation
Segmentation is the process of breaking a sequence of Chinese characters into a sequence of Chinese terms.

Bracketing
Bracketing is similar to segmentation, but it works on English text. Using an English phrase list, our bracketing program identify phrases in a sentence and concatenates words in a phrase with underscores. An English sentence is transformed from a sequence of words to a sequence of terms. For example, the sentence *Speech by President Jiang Zemin at Handover Ceremony* can be bracketed into *Speech_by President_Jiang_Zemin at Handover_Ceremony*.

Tokenization
We define tokenization as the process of finding new Chinese terms and English phrases from the corpus. Because the algorithm for Chinese and English is the same in our experiments, we use "tokenization" to describe this monolingual technique.

3 Previous work

In our previous work (Zhang 2001), the way we improved the segmenter was by augmenting the frequency list with a list of new words found in the corpus. We scanned the corpus for bigrams (two adjacent Chinese characters), trigrams and four-grams. A sliding window and some estimation criteria were used to keep only highly frequent patterns (bigrams, trigrams or four-grams) in memory. The program used these patterns to form the longest possible Chinese term. Mutual information (MI) was used to determine the boundary of the terms.

To match the increased average length of Chinese terms, we performed the equivalent process on the English side of the corpus: scanning the text, using MI to find a list of possible phrases, bracketing the sentence with this phrase list. As described above, the EBMT system we used is a word-based system. The bracketer concatenated the words in a phrase with underscores. Thus EBMT treats such a phrase as a "word" while indexing and translating.

Because the term-finder works monolingually, it may produce excessively long Chinese terms and English phrases which are impossible to match between source language and target language Thus we repeat the procedure of segmenting/bracketing/dictionary-building several times. On each successive iteration, the segmenter and the bracketer are limited to terms and phrases for which the statistical dictionary from the previous iteration contains valid translations.

This pre-processing provided a 12% absolute improvement in coverage of EBMT translations without requiring any additional knowledge resources. Further, the enhanced coverage did, in fact, result in improved translation quality, as verified by human judgements.

The lesson we learned from this previous work is that when we combine words into larger chunks on both sides of the corpus, the possibility of finding larger matches between the source language and the target language increases, which leads to the improvement of the translation quality for EBMT.

The weakness of the algorithm used in previous work was caused by keeping trigrams and four-grams in memory. The size of the trigrams and four-grams was too large; we had to discard some less probable patterns in the process. This led to the potential problem that many new words could not be found.

For HKnews, 5830 terms were found; among them, 4615 are beyond the LDC's original frequency list. In the next 3 sections, we will describe our new approach for bilingual corpus pre-processing.

4 Tokenization Techniques

A monolingual sentence is broken into clauses based on the punctuations in it; we arbitrarily define that no terms can cross punctuations. Unigrams and bigrams of terms are built while reading the corpus. Only adjacent terms inside a clause are considered bigrams.

Collocation measure

For each adjacent pair of terms $(w_1:w_2)$, the collocation is measured by the following formula:

$$Collocation(w_1 : w_2) = \frac{VMI(w_1 : w_2)}{H(w_1) + H(w_2)},$$

where H is the entropy of a word and $VMI(w_1:w_2)$ is a variant of average mutual information:

$$VMI(w_1 : w_2) =$$

$$P(w_1 = 1, w_2 = 1)\log\frac{P(w_1 = 1, w_2 = 1)}{P(w_1 = 1)P(w_2 = 1)}$$

$$+ P(w_1 = 0, w_2 = 0)\log\frac{P(w_1 = 0, w_2 = 0)}{P(w_1 = 0)P(w_2 = 0)}$$

$$- P(w_1 = 1, w_2 = 0)\log\frac{P(w_1 = 1, w_2 = 0)}{P(w_1 = 1)P(w_2 = 0)}$$

$$- P(w_1 = 0, w_2 = 1)\log\frac{P(w_1 = 0, w_2 = 1)}{P(w_1 = 0)P(w_2 = 1)}$$

where $P(w_1=1,w_2=0)$ is the probability of a bigram which has w_1 as its first term and a non-w_2 term as its second term.

We do not use point-wise MI since it does not capture the intuitive notion of collocation very well (Fontenelle et al. 1994). Average MI is the reduction in uncertainty of one variable (w_i=1 or 0) due to knowing about the other, while what we need to know for collocation is the reduction in uncertainty of knowing that one word appears (w_i=1) due to knowing that the other word appears, and vice versa: the reduction in uncertainty of knowing one word should not appear (w_i=0) due to knowing the other word does not appear. Thus, events (w_1=1,w_2=0) and (w_1=0,w_2=1) should be considered as providing negative information to collocation.

The denominator in $collocation(w_1:w_2)$ is a smoothing factor. A high $VMI(w_1:w_2)$ value only shows that w_1 and w_2 have strong tendency to appear together. There is a possibility that one or both of them are highly frequent words, where $H(w_1)$ and/or $H(w_2)$ have high values. Divided by this denominator, collocation values of such word pairs are decreased. But even with this factor, there are still problems such as cross-boundary mistakes caused by the collocation values, as shown in the "Tokenization Procedure" section below.

Segmenting

Once the collocation scores of bigrams in training data are calculated, we use the following algorithm to segment the training data.

For a sentence $w_1\ w_2\ w_3 \ldots w_i\ w_{i+1} \ldots w_n$:

i=1
while(i<n){
  c_i=collocation(w_i:w_{i+1})
  if c_i > $threshold_1$ {
    segment w_i and w_{i+1} as one term;
    if (i<n-1){
      continue=true
      while((i<n-1)&continue)){
        c_{i+1}=collocation(w_{i+1}:w_{i+2})
        if (similarity(c_i, c_{i+1}) >$threshold_2${
          segment w_i, w_{i+1}, w_{i+2} as one term.
          i++}
        else{
          continue=false
          i=i+2
    }}}
    else
      i++
}}

Similarity function: similarity(x,y) is defined as following:

$$similarity\ (x, y) = \begin{cases} x / y, (y \geq x) \\ y / x, (x > y) \end{cases}$$

For example, segment a Chinese sentence 烹煮任何食物以供人食用 */(Pin Yin) peng zhu ren he shi wu yi gong ren shi yong /(English) cook any food that is for people to eat/* into characters first. The histogram of collocation scores for each adjacent pair of terms (characters in this case) is shown in Figure 1. bigram 烹煮 */(Pin Yin) peng zhu/ (English) cook/* has a collocation score 0.58, which is higher than *thresh1=0.1*, so the characters are segmented as one term. The next bigram 煮任 */(Pin Yin) zhu ren/ (English) no meaning/* has a collocation score 0.02. The similarity of these two collocation scores is 0.02/0.58=0.034 which is smaller than *thresh2=0.6*, so only 烹煮 is considered as a term, not 烹煮任 */(Pin Yin) peng zhu ren/ (English) No Meaning/*.
Bigram 以供 */(Pin Yin) yi gong/(English) for the purpose of/*, whose collocation score is 0.19 is higher than *thresh1*, and the collocation score for 供人 is 0.17, *similarity(0.19,0.17)=0.89*, higher than thresh2. If we keep on looking at the following bigrams, we can segment the last five characters as one term 以供人食用 */(Pin Yin) yi gong ren shi yong/(English) for the purpose of being eaten by people/*.

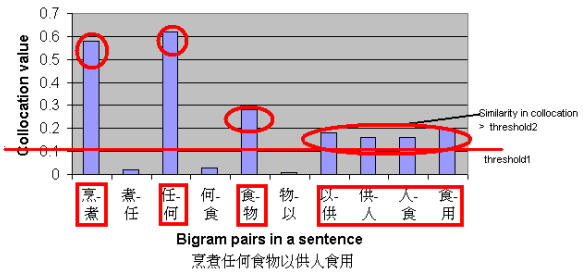


Figure 1: histogram of collocation score

Tokenization Procedure

We first tried to apply this tokenization algorithm directly to the corpus. Although the result was positive, it turned out that purely statistical methods cannot produce a highly accurate segmentation, or it is very difficult to do so. One problem is cross-boundary word segmentation. For instance, word sequence *[X a] [Y]* appears with high frequency in the corpus, where *X*, *Y* are two sets of words. In correct segmentation, *X* and *a* should form a term but in some cases of this pattern, the tokenization program will segment *[a Y]* as a term. Figure 2 shows two examples of such a problem.

To solve this kind of problem, we used the LDC's original frequency list to segment the corpus first. This list can be considered as pre-knowledge with some amount of linguistic information on words. We then ran the tokenization program on this pre-segmented text to cover those words that are not listed. As shown in the results in Section 6, this approach works well.

The process of constructing bigrams/calculating collocation score/ segmenting the training text is repeated several times (in the experiments reported here, 3 iterations) in order to find better tokenization.

1) **Correct segmentation:**

	[X a]	[Y]
	[政务 司]	[司长]
(Pin Yin)	/zheng wu si	si zhang/
(English)	/Dept. of Admin	Secretary of the Dept./

Wrong segmentation:

	[X]	[a Y]
	[政务]	[司 司长]
(Pin Yin)	/zheng wu	si si zhang/
(English)	/Admin.	Dept. Secretary of the Dept./

2) **Correct segmentation:**

	[X a]	[Y]
	[律政 司]	[司长]
(Pin Yin)	/lü zheng si	si zhang/
(English)	/Dept. of Justice	Secretary of the Dept./

Wrong segmentation:

	[X]	[a Y]
	[律政]	[司 司长]
(Pin Yin)	/lü zheng	si si zhang/
(English)	/Justice	Dept. Secretary of the Dept./

Figure 2: Cross-boundary problem

Tokenization for English

The algorithm for tokenizing English is the same as for Chinese except that we used the Porter stemmer (Porter 1980) to stem both the phrase list and the English text before bracketing.

5 Feedback from Statistical Dictionary

Generally speaking, larger units are better for EBMT systems. We have three reasons for this: firstly, larger units (or longer terms) can encapsulate more context in one unit; and it is easier to align source/target text if the unit boundaries matches; thirdly, it can reduce the boundary friction in the target language. But in some cases, over-tokenization in one language may lead to alignment failures because the tokenization program uses only monolingual information to tokenize the text.

A statistical dictionary extracted from the corpus was used to perform sub-sentential alignment in the EBMT system (Brown 1997). We used the results of this dictionary as feedback to adjust tokenization.

The automated dictionary extraction program uses a correspondence table (a two dimensional array counting the collocation between the source and the target language words), which is filtered using a threshold scheme (rather than a measure such as chi-square, mutual information or Dice coefficients). Any word pairs that pass the threshold filter are considered to be translations for the purposes of EBMT alignment. For these pairs (from Chinese term to English word/phrase), we believe the segmentation of source Chinese term and bracketing of target English word/phrase to be proper. Otherwise, if a Chinese term is found with no translation, the segmentation may be wrong. The results from the statistical dictionary are passed to the segmenter and bracketer to re-segment/re-bracket the corpus. This process is repeated several times before the average length of words in the corpus converges.

Figure 3 is the flowchart of the pre-processing stage.
The original corpus was split into Chinese text and English text. The Chinese text was segmented by the LDC segmenter with its original Chinese word frequency list. An English phrase list was used to bracket the English text. The segmented Chinese/bracketed English text was fed into the tokenization program and resulted in the further segmented Chinese/bracketed English text. We aligned these two files to get the bilingual text, from which the statistical dictionary was then built. We combined the Chinese terms from the dictionary with the Chinese words from the glossary to make a new Chinese word frequency list. All Chinese word entries in this list had at least one English translation. The same process was applied to the English side to get a new English phrase list. These two new lists were then used by the segmenter/bracketer to re-segment/re-bracket the original corpus for the next iteration.

The final result of the corpus pre-processing is the aligned bilingual corpus, a statistical dictionary and an augmented Chinese word frequency list, which is used to segment the Chinese sentences in the test-set.

6 Results

The evaluation of tokenization is based on two measures: the average length of the words in the corpus, and the size of the statistical dictionary.

The evaluation was done on HKnews. The original Chinese word frequency list contains 44,404 entries. Table 2 shows the average length (number of Chinese characters) of Chinese terms in segmented text.

Number of Chinese characters per term	LDC seg. with Orig List	Plus new words found by prev. work	Tokenization Prog. (before feedback)
Avg. Len/token	2.21	2.61	5.58
Avg. Len/type	1.44	1.63	2.49

Table 2: Average length of Chinese terms

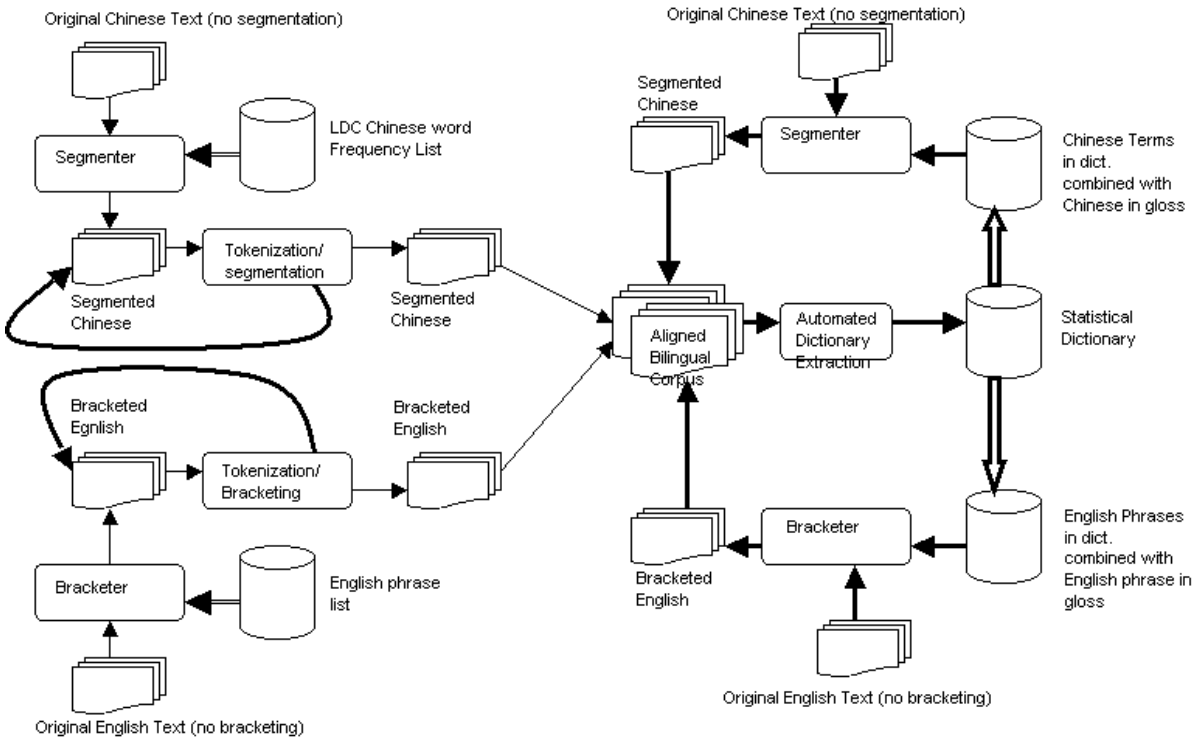


Figure 3: Flowchart of corpus pre-processing

The phrase list from the glossary contains 12,722 entries. Table 3 shows the average length (number of English words) of English terms in bracketed text.

Number of English words per phrase	Using phrase list from gloss	Tokenization Program (before feedback)
Avg. Len/token	1.15	1.95
Avg. Len/type	1.06	1.37

Table 3: Average length of English terms

After the feedback from the statistical dictionary was used, the average length decreased, but the value was still much higher than the average length when only the original list was used. The results are shown in Table 4.

Length	Using Orig. List	After Tok. (Iter.1)	After Feedback (Iter.2)	Afte FeedBack (Iter.3)
Chn /type	2.21	5.58	3.58	3.54
Chn/token	1.44	2.49	2.02	2.00
Eng/type	1.15	1.56	1.39	1.38
Eng/token	1.06	1.26	1.17	1.16

Table 4: Difference of term length in the pre-processing

The main purpose of corpus pre-processing is to increase the quality of alignment, or in other words, the probability of matching between bilingual sentences. This can be evaluated in the size of the statistical dictionary. The more source words found with translations, the better the alignment can be. Table 5 shows our evaluation results.

	Using Orig. List	After Tok. (Iter.1)	After Feedback (Iter.2)	After Feedback (Iter.3)
Chinese Terms in Dict.	8,765	12,821	11,575	11,618
English Phrases in Dict.	0	21,123	13,224	13,109

Table 5: Entries in statistical dictionary

7 Conclusions and Future Work

As seen in Table 4, the pre-processing on the corpus described here increased the average length of Chinese terms about 60%, and 10% for English. The statistical dictionary gained about a 30% increase in coverage. Further, enhanced EBMT coverage does, in fact, result in improved translations, as shown in our previous work (Zhang 2001). Manual judgement of this work is still pending at this time.

We will conduct further research on using feedback from the statistical dictionary to combine or split the current segmentation/bracketing. More information from the

bilingual corpus can be used to guide the monolingual tokenization process.

We have not yet taken full advantage of the features we have developed for the EBMT system. We intend to test automatic creation of equivalence classes from the training corpus (Brown 2000) and named-entity tagging in conjunction with the improvements reported herein.

8 Acknowledgements

We would like to thank Erik Peterson, Stephan Vogel, Yiming Yang and Jie Yang for their comments on this paper.

References

Ralf D. Brown. 1996. Example-Based Machine Translation in the PanGloss System. In *Proceedings of the Sixteenth International Conference on Computational Linguistics*, Pages 169-174, Copenhagen, Denmark. http://www.cs.cmu.edu/~ralf/papers.html

Ralf D. Brown. 1997. "Automated Dictionary Extraction for ‘‘Knowledge-Free” Example-Based Translation". In *Proceedings of the Seventh International Conference on Theoretical and Methodological Issues in Machine Translation*, p. 111-118. Santa Fe, July 23-25, 1997

Ralf D. Brown. 1999. Adding Linguistic Knowledge to a Lexical Example-Based Translation System. In *Proceedings of the Eighth International Conferences on Theoretical and Methodological Issues in Machine Transaltion (TMI-99)*, pages 22-32, Chester, England, August. http://www.cs.cmu.edu/~ralf/papers.html

Ralf D. Brown. 2000. Automated Generalization of Translation Examples. In *Proceedings of the Eighteenth International Conferences on Computational Linguistics (COLING-2000)*, pages 125-131

FDMC. 1986. Xiandai Hanyu Pinlu Cidian (Frequency Dictionary of Modern Chinese). Beijing Language Institute Press

Christopher Hogan and Robert E. Frederking. 1998. An Evaluation of the Multi-engine MT Architecture. In *Machine Translation and the Information Soup: Proceedings of the Third Conference of the Association for Machine Translation in Americas (AMTA ’98)*, volume 1529 of *Lecture Notes in Artificial Intelligence*, pages 113-123. Springer-Verlag, Berlin, October.

Porter, M.F. 1980. An algorithm for suffix stripping. Program 14:130-137.

S. Sato and Makato Nagao. Towards Memory Based Translation. In Proceedings from the 13th International Conference on Computational Linguistics (COLING-90), pp. 247-252, Helsinki, 1990.

Fontenelle, Thierry, Walter Brüls, Luc Thomas, Tom Vanallemeersch, and Jacques Jansen. 1994. DECIDE, MLAP-Project 93-19, deliverable D-1a: survey of collocation extraction tools. Technical report, University of Liege, Liege, Belgium

Ying Zhang, Ralf D. Brown, and Robert E. Frederking, 2001, “Adapting an Example-Based Translation System to Chinese”. In *Proceedings of Human Language Technology Conference*, 2001 (to appear)