Statistical Approaches to Patent Translation for PatentMT-Experiments with Various Settings of Training Data

Yuen-Hsien Tseng
National Taiwan Normal University
No.162, Sec. 1, Heping East Road., Taipei, Taiwan (ROC)
samtseng@ntnu.edu.tw

Chao-Lin Liu, Chia-Chi Tsai, Jui-Ping Wang, Yi-Hsuan Chuang
National Chengchi University
NO.64, Sec.2, ZhiNan Rd., Wenshan, Taipei City 11605,Taiwan (R.O.C)
chaolin@nccu.edu.tw

James Jeng
WebGenie Information Ltd.
No.207-1, Sec. 3, Beisin Rd., Sindian, Taipei, Taiwan (ROC)
james@webgenie.com.tw

ABSTRACT
This paper describes our experiments and results in the NTCIR-9 Chinese-to-English Patent Translation Task. A series of open source software were integrated to build a statistical machine translation model for the task. Various Chinese segmentation, additional resources, and training corpus preprocessing were then tried based on this model. As a result, more than 20 experiments were conducted to compare the translation performance. Our current results show that 1) consistent segmentation between the training and testing data is important to maintain the performance; 2) sufficient number of good quality bilingual training sentences is more helpful than additional bilingual dictionaries; and 3) the translation effectiveness in BLEU values doubles as the number of bilingual training sentences at the level of 100,000 doubles.

Categories and Subject Descriptors
I.2.7 [**Natural Language Processing**]: Machine Translation

General Terms
Documentation, Performance, Experimentation, Languages.

Keywords
Chinese segmentation, language modeling, training corpus.

Team Name: [NCW]

Subtasks/Languages: [Chinese-to-English]

External Resources Used: [Bilingual Lexicons]

1. INTRODUCTION
In recent decades, statistical translation model (STM) has been a very popular approach to machine translation. It is mathematically sound and conceptually simple. With proper implementation, it can adapt to different languages and domains with ease and achieve impressive effectiveness while requiring far less effort to build than traditional approaches such as rule-based machine translation.

Mathematically, the statistical translation model can be described by the following formula:

$$e^* = \underset{e}{\operatorname{argmax}} P(e \mid c) = \underset{e}{\operatorname{argmax}} P(e)P(c \mid e)$$

The above formula shows that to translate, for example, a Chinese sentence c (the source language) into an English sentence e (the target language), we compute the most likely e given c, which is broken down to compute the largest combined probabilities: one probability $P(c|e)$ is called translation model, the other $P(e)$ is called language model.

Although there are quite a few subtle details involved in the above mathematics, Figure 1 demonstrates a conceptual example of how this formula is applied. Given a bilingual corpus with aligned sentences, $P(c|e)$ is built before translation by breaking down the aligned sentences into aligned words, as demonstrated by the term pairs in the middle table. Likewise, given a monolingual corpus, $P(e)$ is built by breaking down the monolingual sentences into short fragments, such as words, phrases, or n-grams, as demonstrated by the colored words in the monolingual corpus. When translating a sentence c, each of the Chinese words in c is translated into all possible English words, each with a translation probability. All the translated English words from all the Chinese words are then combined back to form a word sequence that mostly likes a valid English sentence. The model then chooses a combined word sequence with highest probability as an output translation (such as the first one in the right most table).

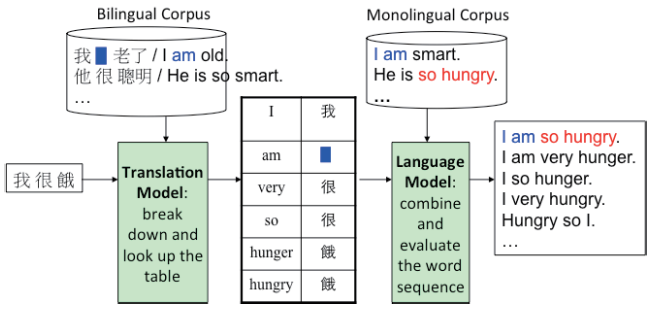

Figure 1. An example to show a statistical translation process.

Although the STM is conceptually simple, as depicted in Figure 1, it has the following difficulties to be overcome:

1) The sentences in the source and the target language need to be segmented into smaller fragments (such as words or phrases) in order for the breakdown to be taken during the above model construction and translation processes. The word segmentation problem can be difficult for some languages such as Chinese, since there is no word boundary

in the Chinese texts. In addition, new words can be easily coined (such as those for abbreviations, acronyms, transliterations, or domain terminology) by combining any existing Chinese characters and words in any texts, which worsens the word segmentation problem. Although this problem is not unique to the STM, the accuracy of the word segmentation and the bilingual word alignment that follows definitely affects the STM performance.

2) The longer the Chinese sentence to be translated, the more the Chinese words there are, and thus the more the translated English words resulted. In turn, this causes many English word sequence combinations to be tried and evaluated. As a result, more errors may exist in the translated word sequence, due to the data sparseness problem (word combination that does not exist in the training data would occur more often than the case for short sentences).

3) There needs a bilingual corpus with aligned sentences to build the translation model. The larger the bilingual corpus, the less the problem of data sparseness. However, high-quality large bilingual corpora are costly to obtain. So how large is enough for the bilingual corpus (e.g., the size/effectiveness ratio) is a question worth of answering.

From the above analysis, it can be seen that the larger the bilingual corpus, the better the translation; and the shorter the source sentence, the easier and more accurate to translate it into a target sentence. Also, segmentation plays an important role in STM. These three factors guide our experiments to explore the Chinese-English patent translation subtask in NTCIR-9 [1], instead of focusing on the core statistical translation techniques that are nowadays relatively accessible through open source STM systems, such as Moses. Therefore, we run experiments based on Moses [2] to explore the effects of various segmentation modules, training data sizes, and additional resources. Details of the resources and tools we used are described below.

2. RESOURCES and PREPROCESSING

2.1 Chinese Segmentation Tools

Before applying the Chinese-English sentence pairs to train the statistical machine translator, the Chinese sentences need to be segmented into word sequences like English for the translator to learn the bilingual word pairs to build the translation model. We employed two publicly available Chinese segmentation tools in this process. The first one is implemented by the Stanford NLP group [3] and the second one by LingPipe [4]. We will refer to these tools for short as Stanford and LingPipe, respectively. We trained Moses with the segmented Chinese and their English translations, and tested the trained models after-wards.

2.2 Chinese Dictionaries

In addition to the 1 million parallel Chinese-English sentence pairs provided by the workshop organizers, we employed two sets of professional Chinese-English dictionaries. The two sets both originated from National Academy for Educational Research [5] in Taiwan. The original data consisted of over 100 large Excel files, totaling 167M in data size. The format of the Chinese-English term pairs in these files is designed for human inspection. That means the terms are usually appended or prefixed with various punctuations and modifiers to signify their usage in different contexts and domains.

In the first set, we made efforts to extract those that are less ambiguous in format, which amounts to about 840,000 term pairs. Besides, terms pairs extracted from about 500,000 bilingual titles downloaded from Taiwan Intellectual Property Office (TWIPO) were added [6]. This yields about 1,355,000 term pairs in our first set.

The second set also originated from the Excel files. In a small experiment, we found that the term pairs extracted contained many terms that should not be considered as professional terms. They were ordinary terms. Hence, we removed those term pairs that were included in either E-HowNet [7] or WordNet [8]. In the end, we had about 690,000 term pairs in the second set.

To distinguish these two sets, we will refer to the **first set of dictionary as Dic1355** and **the second set Dic690** based on the numbers of terms they have. The major difference between these two sets is that Dic1355 contains both patent-relevant terms and many non-professional terms. All the traditional Chinese terms were converted to simplified ones characters by characters. Therefore, the terms in these two sets may not reflect the terms used in the given training and testing corpus from Hong-Kong.

2.3 Training Corpus Preprocessing

When we received the 1 million Chinese-English sentence pairs for training, we analyzed the lengths, sentence segmentations, and quality of the sentences. We found that the sentences were not segmented with perfect ending: some ended with semicolons, and some ended with parentheses. In addition, some sentences were extremely long, making the processing of these sentences quite challenging on our machines. Furthermore, quite a few sentences differ in only a few Chinese characters or English words. We doubt that these large groups of similar sentences or duplicates may bias the trained model. We then chose to use only a portion of the training corpus. Two kinds of preprocessing were adopted that result in two set of training corpora.

In the first set, the original 1 million sentence pairs were treated as 1 million pairs of parallel paragraphs. Each such paragraph pair was segmented into shorter sentences and re-aligned by the sentence aligner we developed based on the idea of Champollion [9]. Like Champollion, our sentence aligner assigned probabilistic scores to the aligned sentence pairs. We chose only those one-to-one pairs with relatively higher scores to obtain 1,140,000 pairs of short sentences. We manually sampled some pairs, and found that the alignment results satisfactory. With a bit more analysis, we found that these 1,140,000 short sentence pairs belonged to just about 330 thousand pairs in the original 1 million sentence pairs.

The second set of training corpus was derived by removing those long and similar pairs from the original 1 million sentence pairs. That is, sentence pairs with their Chinese longer than n Chinese characters (n=30) were removed and pairs with a between-sentence similarity larger than a threshold (0.85) were also removed. This preprocessing results in about 220,000 pairs of sentences. They were then subjected to a Chinese segmenter developed by WebGenie[1] [10] to result in a segmented bilingual training corpus.

To distinguish these two sets, we will refer to the **first set of training corpus as C1140** and **the second set C220**, based on their corpus sizes. Besides the corpus size, the major differences

[1] The WebGenie's Chinese segmenter is customized for the patents from TWIPO, which uses traditional Chinese. For the simplified Chinese in this task, we try other segmenters customized for this task, as described in the Section 3.1.

between these two sets lie in that the C220 may contain far less duplicate (similar) sentence pairs. For the convenience of later discussion, we will further refer to the **Chinese portions** of C1140 and C220 as **C1140_C**, **C220_C**, respectively, and the **English portions** of C1140 and C220 as **C1140_E** and **C220_E**, respectively.

3. TRAINING

We explored several combinations of the tools for Chinese segmentation and the training corpora in the experiments, as described below.

3.1 Training the Chinese Segmenters

The LingPipe tool needs segmented Chinese to train its model for segmentation, while the Stanford tool can be trained with segmented corpus or be used directly with its built-in models. Hence we use the Chinese portion in C220 to train LingPipe and Stanford. In addition, LingPipe allowed us to provide existing dictionaries when we trained its segmentation models. Since we have two dictionaries, there were four possible ways to train LingPipe models, depending on whether or not we used Dic690 and Dic1335.

Table 1 summarizes five ways that we used to create segmentation models. The right-most column shows the codes that we refer to the resulting segmenters.

Table 1. Five trained segmentation models

Tool	Corpus + Dictionary	Segmenter ID
LingPipe	C220	S1
LingPipe	C220 + Dic690	S2
LingPipe	C220 + Dic1355	S3
LingPipe	C220 + Dic1355 + Dic690	S4
Stanford	C220	S5

3.2 Training Translators with C220

The trained segmenters were then used to re-segment the Chinese texts to train the translation model of Moses. We explained two different ways to use C220 for training in this subsection.

Recall that the **Chinese portion in C220** (**denoted as C220_C**) has already been segmented (by WebGenie for initial segmentation to train LingPipe). Hence, we could directly use the original C220 to train Moses. However, to get more accurate segmentation for the testing data, the simplified Chinese segmenters listed in Table 1 were used (instead of the WebGenie segmenter which is optimized for traditional Chinese) to segment the testing data. This leads to two different ways to segment Chinese in a workflow. That is, the training data and testing data were segmented by inconsistent segmenters.

For this reason, we ran another set of experiments. In this second set, we re-segmented the Chinese portion in C220 (i.e., C220_C) with the trained segmenters, and used the re-segmented Chinese to train Moses. With this approach, both the training and the testing data would be segmented with the same tool. We had expected that the resulting quality of translation would be better, and this expectation was supported in later experiments.

Table 2. Six sets of segmented Chinese

Code	Description
C220_C	Original segmented Chinese in C220
C220_C1	C220_C re-segmented by segmenter S1
C220_C2	C220_C re-segmented by segmenter S2
C220_C3	C220_C re-segmented by segmenter S3
C220_C4	C220_C re-segmented by segmenter S4
C220_C5	C220_C re-segmented by segmenter S5

Table 2 summarizes how we obtained the Chinese text to train Moses. The left column shows the codes for the sets of segmented Chinese. Using these six sets of segmented Chinese in C220 and the corresponding English portion of C220 (i.e., C220_E), we trained Moses and obtained six translators. Table 3 lists the translators, where the left column shows the codes U0-U5 for the resulting translators.

Table 3. Six translators for C220

Code	Description
U0	Train Moses with C220_C and C220_E
U1	Train Moses with C220_C1 and C220_E
U2	Train Moses with C220_C2 and C220_E
U3	Train Moses with C220_C3 and C220_E
U4	Train Moses with C220_C4 and C220_E
U5	Train Moses with C220_C5 and C220_E

3.3 Training Translators with C1140

The Chinese in C1140 was not segmented, so it was not as complicated to evaluate it as we did for evaluating C220. We segmented the Chinese portion in C1140 (i.e., C1140_C) with segmenters S1-S5 in Table 1 to create training data C1140_C1, C1140_C2, C1140_C3, C1140_C4, and C1140_C5, respectively. Analogous to how we created translators V1-V5 in Table 3, we used **C1140_Ci** (i=1, 2, …, 5)) and **C1140_E** to create five translators **V1**, **V2**, **V3**, **V4**, and **V5**, respectively.

3.4 Training Translators without Training our own Chinese Segmenters

As mentioned in Sub-Section 3.1, the Stanford tool can be used directly based on one of three bundled segmentation models, i.e., *ctb*, *pku*, and *christ6*. With the Chinese portions in both C220 and C1140 being segmented by these three existing models and with the Moses being trained with these resulting Chinese, we have three additional translators **X1**, **X2**, and **X3** for C1140 and additional three **Y1**, **Y2**, and **Y3** for C220 in experiments.

4. TESTING WITH TUNING DATA

4.1 Using Translators U's for C220

We segmented the tuning data with segmenters S1-S5 to obtain testing data T1, T2, T3, T4, and T5. These five sets of Chinese texts were translated by the above translators U's and V's. First T1 to T5 were translated by U1 to U5, respectively. The qualities of the translations are shown in the Z1 to Z5 rows in Table 4. T1 to T5 were also each translated by translator U0, and the results are shown in Z6 to Z10, respectively, in Table 4.

Table 4. Ten experiments with C220

Exp. Code	Translator	NIST	BLEU
Z1	U1	7.391	0.249
Z2	U2	6.783	0.220
Z3	U3	5.891	0.173
Z4	U4	6.195	0.181
Z5	U5	6.215	0.191
Z6	U0	3.311	0.067
Z7	U0	2.881	0.062
Z8	U0	2.764	0.060
Z9	U0	3.051	0.064
Z10	U0	2.928	0.055

4.2 Using Translators V's for C1140

We tested the effectiveness of the V translators with a procedure very similar to that we tested the U translators. We simply translated T1 to T5 with V1 to V5 to acquire the results shown in Z11 to Z15 in Table 5.

Table 5. Five experiments with C1140

Exp. Code	Translator	NIST	BLEU
Z11	V1	5.195	0.149
Z12	V2	4.941	0.147
Z13	V3	6.034	0.192
Z14	V4	6.330	0.200
Z15	V5	6.461	0.205

4.3 Using Translators X's and Y's

We segmented the tuning data with *ctb*, *pku*, and *christ6* models of the Stanford segmenter and translated the results by X1, X2, and X3 for C1140 and by Y1, Y2, and Y3 for C220. Table 6 shows the results in Z16, Z17, Z18, Z16*, Z17*, and Z18*, respectively.

Table 6. Three experiments with C1140

Exp. Code	Segmentation Model	NIST	BLEU
Z16	*Ctb* (for C1140)	7.378	0.241
Z17	*Pku* (for C1140)	7.204	0.233
Z18	*Christ6* (for C1140)	7.382	0.240
Z16*	*Ctb* (for C220)	7.435	0.250
Z17*	*Pku* (for C220)	7.399	0.251
Z18*	*Christ6* (for C220)	7.612	0.260

4.4 A Preliminary Comparison

Among the 21 experiments that we explored in this section, directly using the segmentation models in the Stanford NLP tools to segment Chinese led to better translators than using either C220 or C1140 as segmentation training data. The best BLEU scores were achieved in Z16*, Z17*, and Z18*. Using C220 made us create better translators, as the results in Z1-Z5 and Z11-Z15 suggested. Using different segmenters for training and tuning data is a bad idea, as the results in Z1-Z5 and Z6-Z10 showed.

5. SUBMITTED RESULTS

In Section 4, we used the tuning data in the place of the testing data to evaluate our translators. We translated the actual testing data with the 21 procedures that we discussed in Section 4, but we failed to submit the translation results for the last three procedures (i.e., those that produced Z16*, Z17*, and Z18* in Table 6) before the deadline for submission. Instead, we submitted four more experimental results based on a smaller (but cleaner) training corpus and newly acquired resources to see how they can be helpful.

First, from C220, we used more stringent similarity threshold and sentence length criteria to get 70,000 sentence pairs, which is denoted as **C70** training corpus. Similar to the derivation of Dic1335, we obtained a larger bilingual lexicon after several months of acquiring Dic1335. This larger lexicon is denoted as **Dic1700** because it contains about 1,700,000 Chinese-English term pairs. During the deadlines between the dry run and the formal submission, we obtained 50,000 manually corrected Chinese-English sentence pairs from TWIPO patent documents. This corpus is denoted as **Cm50**. All the above additional resources are originally in traditional Chinese and were converted into simplified Chinese for the translation task. The four additional results, denoted as Z19, Z20, Z21, and Z22 are obtained by training the Moses system with C70 (Z19), C70+Dic1700 (Z20), Dic1700 only (Z21), and C70+Cm50 (Z22).

When we submitted the translation results, we ranked the results based on the BLEU scores of the tuning data. Table 7 shows the correspondence between the experiment codes in Tables 4, 5, 6 and the IDs for our final submission. The evaluation results for the tuning data and the actual testing data are put together in Table 7 for easy comparison.

Table 7. BLUE scores of NCW experiments

NTCIR9 ID	Exp. Code	Tuning	Test
G13-ze-01	Z1	0.249	0.258
G13-ze-02	Z16	0.241	0.242
G13-ze-03	Z18	0.240	0.243
G13-ze-04	Z17	0.233	0.231
G13-ze-05	Z2	0.220	0.234
G13-ze-06	Z15	0.205	0.209
G13-ze-07	Z14	0.200	0.205
G13-ze-08	Z13	0.192	0.215
G13-ze-09	Z5	0.191	0.196
G13-ze-10	Z4	0.181	0.182
G13-ze-11	Z3	0.173	0.192
G13-ze-12	Z11	0.149	0.157
G13-ze-13	Z12	0.147	0.155
G13-ze-14	Z6	0.067	0.065
G13-ze-15	Z9	0.064	0.058
G13-ze-16	Z7	0.062	0.065
G13-ze-17	Z8	0.060	0.063
G13-ze-18	Z10	0.055	0.053
G13-ze-19	N/A	N/A	0.120
G13-ze-20	N/A	N/A	0.103
G13-ze-21	N/A	N/A	0.017
G13-ze-22	N/A	N/A	0.125

The correlation coefficient between the BLEU scores of the tuning data and the test data is very high. For individual test procedures, the differences between the BLEU scores for the tuning and test data are not large either.

Although using the Stanford segmentation models led the best performing translators for the tuning data, using re-segmented C220 and no extra dictionaries to train LingPipe led the performance for the final tests, cf. the Z1 row in Table 7. Our results also show that using only bilingual dictionary (G13-ze-21) perform worse than using only bilingual sentences (G13-ze-19 and runs).

6. CONCLUSIONS

In our experiments, we did not dive into tweaking the open source STM tools for better performance. Instead, we have the following major concern in mind to guide our experiments: Since statistical translation model relies heavily on the quality and quantity of the training data, how the performance is affected by these factors quantitatively is useful information to know. Our results show that, giving a reasonable quality of the training data, the translation effectiveness is somewhat linear to the quantity of the training data, i.e., 0.125 in BLEU for about 120,000 (70,000+50,000) training sentences (G13-ze-22) and 0.258 for about 220,000 of bilingual sentences (In [11], the experiment results from all teams for Japanese-English patent translation also show similar linear relation before performance saturation). So far, large sets of Chinese-English sentence pairs for patent translation are still costly to acquire. Excessive bilingual pairs far more than needed (beyond performance saturation) is considered to be a waste of effort and budget. Knowing the relations between the translation performance and the quantity/quality of the training set allows us to estimate more accurately the cost needed and/or the possible performance that can be achieved. This cost/effect information is practically useful to the organization such as TWIPO when they plan to adopt a statistical machine translation approach for their (traditional) Chinese-English patent translation services.

7. ACKNOWLEDGMENTS

Our work was partially supported by funding from the National Science Council in Taiwan under contracts NSC-99-2221-E-004-007, NSC-100-2221-E-004-014, NSC 97-2628-E-003-003-MY3, and by the project of Aim for the Top University Plan, sponsored by Ministry of Education, Taiwan, R.O.C.

8. REFERENCES

[1] Isao Goto, Bin Lu, Ka Po Chow, Eiichiro Sumita and Benjamin K. Tsou, Overview of the Patent Machine Translation Task at the NTCIR-9 Workshop, NTCIR-9, 2011.

[2] Moses: http://www.statmt.org/Moses/

[3] The Stanford Natural Language Processing Group: http://nlp.stanford.edu/software/segmenter.shtml

[4] LingPipe: http://alias-i.com/lingpipe/

[5] National Academy for Educational Research, Taiwan: http://terms.nict.gov.tw/download_main.php

[6] Yuen-Hsien Tseng, Chao-Lin Liu, Ze-Jing Chuang "Automatic Term Pair Extraction from Bilingual Patent Corpus" The 21st Conference on Computational Linguistics and Speech Processing (ROCLING 2009), p.279-292.

[7] E-HowNet: http://ckip.iis.sinica.edu.tw/taxonomy/taxonomy-edoc.htm

[8] WordNet: http://wordnet.princeton.edu/

[9] X. Ma. Champollion: A robust parallel text sentence aligner, Proc. of the 5th Int'l Conf. of the Language Resources and Evaluation, 489–492, 2006.

[10] WebGenie: http://www.webgenie.com.tw/

[11] Atsushi Fujii, Masao Utiyama, Mikio Yamamoto and Takehito Utsuro, "Overview of the Patent Translation Task at the NTCIR-7 Workshop", Proceedings of NTCIR-7 Workshop Meeting, December 16–19, 2008, Tokyo, Japan, pp.389-400.