The ICT Statistical Machine Translation Systems for IWSLT 2007

Zhongjun He, Haitao Mi, Yang Liu, Deyi Xiong,
Weihua Luo, Yun Huang, Zhixiang Ren, Yajuan Lu, Qun Liu

Key Laboratory of Intelligent Information Processing, Institute of Computing Technology
Chinese Academy of Sciences
No.6 Kexueyuan South Road, Haidian District
P.O. Box 2704, Beijing, China, 100080
{zjhe, htmi, yliu, dyxiong, luoweihua, huangyun, renzhixiang, lvyajuan, liuqun}@ict.ac.cn

Abstract

In this paper, we give an overview of the ICT statistical machine translation systems for the evaluation campaign of the International Workshop on Spoken Language Translation (IWSLT) 2007. In this year's evaluation, we participated in the Chinese-English transcript translation task, and developed three systems based on different techniques: a formally syntax-based system Bruin, an extended phrase-based system Confucius and a linguistically syntax-based system Lynx. We will describe the models of these three systems, and compare their performance in detail. We set Bruin as our primary system, which ranks 2 among the 15 primary results according to the official evaluation results.

1. Introduction

This paper describes the statistical machine translation systems of Institute of Computing Technology, Chinese Academy of Sciences, which are used for the evaluation campaign of the International Workshop on Spoken Language Translation (IWSLT) 2007.

We participated in the Chinese-English transcript translation task and developed three SMT systems based on different techniques for this year's evaluation. The primary system is Bruin, which is a formally syntax-based system. Another one is an extended phrase-based system named Confucius, and the third one is a linguistically syntax-based system named Lynx.

This paper is structured as follows: Section 2 gives an overview of our three SMT systems, Section 3 describes data preparation methods. In Section 4, we will report the experiments and results. Finally, Section 5 gives conclusions.

2. Systems Overview

Our group focused on statistical machine translation since 2004. During the last three years, we have tried various methods and developed three SMT systems: Bruin (a formally syntax-based system), Confucius (an extended phrase-based system) and Lynx (a linguistically syntax-based system). In this section, we will describe the models and algorithms in detail.

2.1. Bruin

Bruin is a formally syntax-based SMT system, which implements the maximum entropy based reordering model on BTG [1] rules.

There are three essential elements in Bruin. The first one is a stochastic BTG, whose rules are weighted using different features in the log-linear form. The second is a MaxEnt-based reordering model predicting the orders between neighbor blocks, whose features are automatically learned from bilingual training data. The last one is a CKY-style chart-based decoder with beam search similar to that of Wu [1].

2.1.1. Translation Model

To complete the decoding procedure, three BTG rules are used to derive the translation:

$$A \xrightarrow{[\,]} (A^1, A^2) \tag{1}$$

$$A \xrightarrow{\langle\,\rangle} (A^1, A^2) \tag{2}$$

$$A \rightarrow (x, y) \tag{3}$$

The lexical rule (3) is used to translate source phrase y into target phrase x and generate a block A. The merging rules (1) and (2) are used to merge two consecutive blocks into a single larger block in the straight or inverted order.

To construct a stochastic BTG, we calculate rule probabilities by the log-linear model. For the two merging rules *straight* and *inverted*, applying them on two consecutive blocks A^1 and A^2 is assigned a probability $Pr^m(A)$

$$Pr^m(A) = \Omega^{\lambda_\Omega} \cdot \triangle_{p_{LM}(A^1,A^2)}^{\lambda_{LM}} \tag{4}$$

where the Ω is the reordering score of block A^1 and A^2, which is calculated by the MaxEnt-based reordering model

described in the Section 2.1.2, λ_Ω is its weight. The $\triangle_{p_{LM}(A^1,A^2)}$ is the increment of the language model score of the two blocks according to their final order, λ_{LM} is its weight.

For the lexical rule, applying it is assigned a probability $Pr^l(A)$:

$$\begin{aligned} Pr^l(A) &= p(x|y)^{\lambda_1} \cdot p(y|x)^{\lambda_2} \cdot p_{lex}(x|y)^{\lambda_3} \\ &\cdot p_{lex}(y|x)^{\lambda_4} \cdot exp(1)^{\lambda_5} \cdot exp(|x|)^{\lambda_6} \\ &\cdot p_{LM}^{\lambda_{LM}}(x) \end{aligned} \quad (5)$$

where $p(\cdot)$ are the phrase translation probabilities in both directions, $p_{lex}(\cdot)$ are the lexical translation probabilities in both directions, and $exp(1)$ and $exp(|x|)$ are the phrase penalty and word penalty, respectively.

The feature weights λs are tuned to maximize the BLEU score on the development set, using minimum-error-rate training [2].

2.1.2. MaxEnt-based Reordering Model

The MaxEnt-based Reordering Model (MRM) is defined on the two consecutive blocks A^1 and A^2 together with their order $o \in \{straight, inverted\}$ according to the maximum entropy framework.

$$\Omega = p_\theta(o|A^1, A^2) = \frac{exp(\sum_i \theta_i h_i(o, A^1, A^2))}{\sum_o exp(\sum_i \theta_i h_i(o, A^1, A^2))} \quad (6)$$

where the functions $h_i \in \{0, 1\}$ are model features and the θ_i are the weights.

There are three steps to train an MRM. We firstly extract reordering examples from the training corpus, then generate features from these examples and finally estimate the feature weights.

Extract reordering examlpes: A *reordering example* is defined as a triple of (o, A^1, A^2), where A^1 and A^2 are two neighbor blocks and o is the order between them. Any algorithms extracting bilingual phrases can be easily modified to extract reordering examples. Additionally, There are two points worth of mentioning:

1. To extract all useful examples, there is no length limit over blocks compared with extracting bilingual phrase.
2. To keep the number of reordering examples acceptable when enumerating all combinations of neighbor blocks, it is a good way to extract smallest blocks with *straight* order while largest blocks with *inverted* order.

Generate reordering features: It is found that the boundary words of the blocks keep information for their movements/reorderings [8]. Here we define our lexical features on the right boundary words (i.e., the first or last words of blocks). We only use the lexical feature of the last words of blocks in Bruin.

Estimate feature weights: This can be done by off-the-shelf MaxEnt toolkits. We use the toolkit implemented by Zhang [1] to tune the feature weights. We set iteration number to 100 and Gaussian prior to 1 to avoid overfitting.

2.1.3. Decoder

The decoder is built upon the CKY chart-based algorithm. To avoid exploring all derivations before the best final derivation is found, we adopt a beam search strategy to prune some bad derivations.

For more details, please refer to [8].

2.2. Confucius

Confucius is an extended phrase-based SMT system, which uses a phrase-based similarity translation model.

Given a source sentence f, the basic phrase-based system [5] first enumerates all possible source phrases $\widetilde{f}$, and searches the bilingual phrase table to find all candidate phrase translations with exact matching in source side. Then, decoding algorithm can be applied to find the best translation $\hat{e}$ according to the following formula:

$$\hat{e} = \underset{e}{\operatorname{argmax}}\, Pr(e|f) \quad (7)$$

However, the exact matching strategy can only select translations for source phrases which have translations in the phrase table, but can not do that for the unseen phrases. Thus, such a method resulting in a data sparseness problem.

The phrase-based similarity model can overcome this problem. In our model, the target translations for unseen source phrases can be derived from the similar phrase pairs in the phrase table. Given a source phrase f_1^J, the model first searches for the most similar phrase pair $(f'^J_1, e'^I_1, \widetilde{a})$ by computing similarity in source side, and then uses the similar phrase pair to construct a new phrase pair by replacing different source words and their corresponding target words. See Figure 1 for illustration.

Given two phrases $\widetilde{f}_1^J = c_1, c_2, \ldots, c_J$, $\widetilde{f'}_1^J = w_1, w_2, \ldots, w_J$, we compute phrase similarity as follows:

$$SIM(\widetilde{f}_1^J, \widetilde{f'}_1^J) = \frac{\sum_{j=1}^J \delta_{c_j w_j}}{J} \quad (8)$$

where,

$$\delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{else} \end{cases} \quad (9)$$

For example,

$$\begin{aligned} SIM(&\text{“全省 出口 总值 的 25.5%”}, \\ &\text{“全市 出口 总值 的 半数”}) \\ &= 3/5 = 0.6 \end{aligned}$$

The similarity equals to 1 means that the two phrases are exactly the same, and 0 means totally different. In our

[1] See http://homepages.inf.ed.ac.uk/s0450736/maxent_toolkit.html.

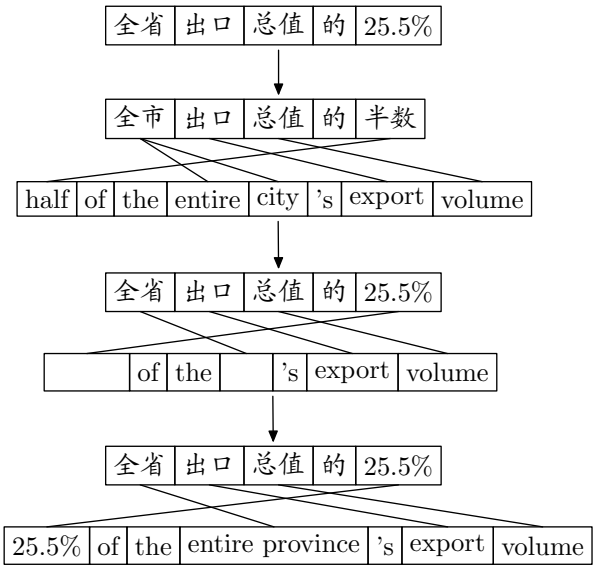

Figure 1: Constructing new phrase pair for a source phrase

experiments, we set the similarity threshold to 0.6, which means that only the similarity above 0.6 can a phrase pair $(f'^{J}_{1}, e'^{I}_{1}, \widetilde{a})$ be used to create translations for f^{J}_{1}.

Please note that usually there are many possible translations for a source phrase in the phrase table, thus more than one phrase pairs can be constructed for a certain source phrase by choosing different translations. For example, the model can produce another phrase pair 全省 出口 总值 的 *25.5%, 25.5% of the total province 's export volume*) by selecting another translations *"total province"* for "全省" in Figure 1.

Following [4], we use 4 probabilities to describe how well a source phrase $\widetilde{f}$ is aligned to a target phrase $\widetilde{e}$: $p(\widetilde{f}|\widetilde{e})$, $p(\widetilde{e}|\widetilde{f})$, $p_w(\widetilde{f}|\widetilde{e})$, $p_w(\widetilde{e}|\widetilde{f})$. Thus, we should score a newly constructed phrase pair ($\widetilde{f}'$,$\widetilde{e}'$) derived from ($\widetilde{f}$,$\widetilde{e}$,$\widetilde{a}$). $p(\widetilde{f}'|\widetilde{e}')$ and $p(\widetilde{e}'|\widetilde{f}')$ are inherited from the father phrase pair, that is:

$$p(\widetilde{f}'|\widetilde{e}') = p(\widetilde{f}|\widetilde{e}) \tag{10}$$

and

$$p(\widetilde{e}'|\widetilde{f}') = p(\widetilde{e}|\widetilde{f}) \tag{11}$$

The lexical weight can be computed according to the substitution of source and target words. Suppose $S\{(f,e)\}$ is a pair set in ($\widetilde{f}$,$\widetilde{e}$,$\widetilde{a}$) which was replaced by $S\{(f',e')\}$ to create the new phrase pair ($\widetilde{f}'$,$\widetilde{e}'$,$\widetilde{a}$), the lexical weight is computed by:

$$p_w(\widetilde{f}'|\widetilde{e}', \widetilde{a}) = \frac{p_w(\widetilde{f}'|\widetilde{e}', \widetilde{a}) \times \prod_{(f',e')\in S\{(f',e')\}} p_w(f'|e')}{\prod_{(f,e)\in S\{(f,e)\}} p_w(f|e)} \tag{12}$$

To train the phrase-based similarity model, we extract the phrase pairs from the word aligned bilingual corpus, which is analogous to the other phrase-based systems. The difference is that we keep the word alignment of the phrase pairs for constructing new phrase pairs, such as (全市 出口 总值 的 半数 ||| half of the entire city 's export volume ||| 1-4 1-5 2-7 3-8 5-1). We use the beam search algorithm for decoding.

2.3. Lynx

Lynx is a decoder based on tree-to-string alignment template (TAT), which describes the alignment between a source parse tree and a target string. A TAT is capable of generating both terminals and non-terminals and performing reordering at both low and high levels. The TAT-based model is linguistically syntax-based because TATs are extracted automatically from word-alignment, source side parsed parallel texts. To translate a source sentence, we first employ a parser to produce a source parse tree and then apply TATs to transform the tree into a target string. More details can be found in [7]. We used seven feature functions analogous to default feature set of Pharaoh [5]. The Chinese sentences were parsed with a Chinese parser developed by Xiong et al. (2005). The parser was trained on articles 1-270 of Penn Chinese Treebank version 1.0 and achieved 79.4% (F1 measure) as well as a 4.4% relative decrease in error rate.

3. Data Preparation

Although the models and algorithms for our three systems are quite different, they share some of the data preparation methods in our experiments. We describes them in this section.

3.1. Data Preprocessing

Preprocessing is the first step we do for the training data. In our experiments, the following steps are performed for training corpus:

- Tokenization: We do tokenization for both Chinese and English. This step transforms Chinese characters into Chinese words, and separates punctuation from words in both Chinese and English sentence;
- True case mapping: This step is only for English. We check the beginning words of each English sentences in the training corpus, if its lowercase version occurs more often, then we map the uppercase to its lowercase;
- SBC case to DBC case: This step is only for Chinese. Numbers and English words often occurs in SBC case in Chinese, such as "１ ２ ３", "Ａ Ｂ Ｃ", which are replaced by their DBC case "123", "ABC" in this step.

3.2. Word Alignment

To get the word aligned corpus, we perform the following two steps:

- Run GIZA++ [9] to IBM model 4 in both translation directions to get a initial word alignment

Table 1: *Training Data List.*

Names	Description	Sentence Pairs	Chinese Words	English Words
IWSLT2007	Training data provided by IWSLT 2007	39,943	354k	378k
LDC2002L27	Chinese-English Translation Lexicon Version 3.0	79,369	79k	123k
2004-863-008	Dialog corpus from ChineseLDC	51,694	486k	509k
CLDC-LAC-2003-004	Chinese-English Sentence aligned Bilingual Corpus from ChineseLDC	199,702	2.7M	3.1M
CLDC-LAC-2003-006	Chinese-English Sentence aligned Bilingual Corpus from ChineseLDC	299,952	4.5M	4.7M

- Apply "grow-diag-final" method [4] to refine it.

3.3. Phrase Extraction

Generally, in the phrase-based systems, any consecutive words sequence can be seen as phrases, and not necessarily in any syntactic theory. Experiments show that bilingual phrases are very useful even for linguistically syntax-based SMT system. Thus, phrases are used in all our three systems.

We consider bilingual phrase as a pair of source and target words sequence, with the following constraints:

1. The words should be consecutive in both source and target sentences, without any gap;

2. The word alignment of the bilingual phrase should consist with the sentence alignment, which means that the words in a phrase can only be aligned to each other, and not to any other words outside.

In our experiments, we find that some phrases are more likely to appear at the beginning of sentences, while some are more likely to appear at the end. Usually, the positions of these phrases are fixed in a sentence and should not be reordered. In order to learn the beginning and ending phrases information, we add two special tags "<s>" "</s>" to the word aligned sentence. Make sure that the beginning tag in source sentence is aligned to the beginning tag in target sentence, and the same to the ending tag. see figure 2 for illustration.

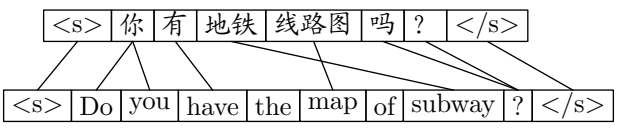

Figure 2: Word aligned sentence with beginning/ending tags

When doing phrase extraction, the beginning and ending tags are extracted together with other words, such as "<s> 你 有 ||| <s> Do you have", "吗 ? </s> ||| ? </s>".

During decoding, we first add the beginning/ending tags to a given sentence, and then apply the search algorithm. Finally, the tags are eliminated from the target sentence.

4. Experiments

In this year's evaluation, we participated in the Chinese-English transcript translation task. This section describes our experiments and results in detail.

4.1. Corpus

We use 670k sentence pairs with 8.3M Chinese words and 8.8M English words as the training data. Table 1 shows the details. The first corpus is provided by the IWSLT 2007; the second corpus is a dictionary released by LDC; the last three corpora are come from Chinese LDC [2], whose domain includes travel, trade, traffic, economy etc. Additionally, we use Chinese-English Name Entity Lists v1.0(LDC2005T34) for translating name entities.

A 4-gram language model is trained on the training corpus using SRI language modeling toolkit [10].

We do our experiments on the test set of the IWSLT 2006 and IWSLT 2007 evaluation. Please note that we resegment the Chinese sentence with ICTCLAS [3]. The corpus statistics are shown in Table 2 . We can see that in this year's evaluation, both the running words and vocabulary are much less than last year, and the average sentence length is only 6.7.

4.2. Results

We carried our experiments on the IWSLT 2006 development set and test set on two conditions: small data and large data.

[2] http://www.chineseldc.org/EN/index.htm
[3] http://www.nlp.org.cn/project/project.php?proj_id=6

Table 2: *Corpus statistics of the IWSLT 2006 and 2007 development and test set.*

	Chinese	English
IWSLT'06-dev Sentences	489	
Running Words	5983	45720
Vocabulary	1139	2150
IWSLT'06-tst Sentences	500	
Running Words	6359	51227
Vocabulary	1331	2346
IWSLT'07-tst Sentences	489	
Running Words	3297	22574
Vocabulary	879	1527

Table 3: *The results of our systems on IWSLT 2006 development set and test set under small data and large data conditions. Please note that Lynx don't run on large data condition.*

Condition	System Name	IWSLT'06-dev	IWSLT'06-tst
Small Data	Bruin	0.1756	0.1731
	Confucius	0.1724	0.1700
	Lynx	0.1681	0.1667
Large Data	Bruin	0.2114	0.2283
	Confucius	0.2115	0.2042
	Lynx	-	-

For the small data condition, we only use the training data released by the IWSLT 2007, and for the large data condition, we use all the training data in Table 1. The results are shown in Table 3. From the table, we can see that the large data greatly improves systems' output quality, since the domain of the corpus is closely to the development and test set.

For this year's evaluation, we use all the training data to train our SMT models, and run all the three systems on the test set. Table 4 shows the official evaluation results. Our primary result produced by Bruin ranked 2 among all the 15 primary results.

4.3. Discussion

We developed three systems based on different techniques for this year's evaluation. However, their performances are surprising. There is about 9-BLEU-point difference between Bruin and Confucius, and about 20-BLEU-point between Bruin and Lynx. However, the experiments on the IWSLT 2006 show that the BLEU scores of the systems are close to each other. Here are some reasons:

Table 4: *The results on IWSLT 2007 test set.*

System Name	IWSLT'07-tst
Bruin	0.3750
Confucius	0.2802
Lynx	0.1777

Training Corpus: Besides the training data provided by IWSLT 2007, Lynx uses about 5M sentence pairs newswire data released by LDC to train the model for this years' evaluation. We think that the more data we use, the better result it will be. However, the experiment results show that the larger data failed to bring any improvement. This is because that the domain is quite different between the training data and the test data. The model trained on newswire data is not fit to translate the dialogs. More seriously, the large amount irrelevant data is harmful to the model.

The Parser: As mentioned in Section 2.3, the input sentences for Lynx are parsed by a Chinese parser trained on Penn Chinese Treebank, whose domain is quite different from the IWSLT test set. As a result, the input sentences may have many parsing errors. This is another main reason why the BLEU score of Lynx is very low.

The Phrase-based similarity model: Confucius uses the phrase-based similarity model to overcome data sparseness, in which unseen phrases can be translated according to the similar phrase pairs in the phrase table. In our experiments, comparing to the baseline system which uses exact matching when selecting translation options, our model can achieves an absolute improvement of 1.1% on BLEU-4. We find that the extended phrases are very useful, and will use the phrase-based similarity model in Bruin in future.

The Reordering Model: Bruin implements a MaxEnt-based reordering model, which can do long distance word reordering. However, Confucius only perform monotone search. The difference on BLEU score is more distinct for this year's test set. There are two reasons for such a different result: Firstly, as shown in Table 2, the sentences in this year's test set are much shorter than last year's, only 6.7 words per sentence on average, while the length is 12.7 in last year. As the sentences are shorter, the word reordering can only occur in a shorter distance, thus the word reordering model works well. In other words, the word reordering model does better job for short sentence than long sentence. Secondly, the test set of this year contains punctuation marks, while last year's test set doesn't. The punctuation is very useful for word reordering. For example, if the last word of a sentence is "?", it may suggest that the word order should be changed between the interrogative and other component of the sentence, since usually the interrogative is at the end in Chinese while at the beginning in English.

See the following example:

[source] 到 洛杉矶 需要 多 长 时间 ？
[Bruin] How long does it take to get to Los Angeles ?
[Confucius] To Los Angeles how long does it take ?
[Lynx] Los Angeles need a long time ?

In the example, Bruin changes the order between "到 洛杉矶" and "需要 多 长 时间", which produces the correct translation. Unfortunately, Confucius and Lynx cannot do the phrase reordering.

5. Conclusions

In this paper, we describes the ICT statistical machine translation systems for the evaluation campaign of the International Workshop on Spoken Language Translation (IWSLT) 2007. We have developed three different kinds of SMT systems, the formally syntax-based system Bruin, the extended phrase-based system Confucius and the linguistically syntax-based system Lynx. We also give a detail discussion on the performance of these three systems. For the evaluation, we try to do result reranking and system combination, but they don't work well. We will do this in the future.

6. Acknowledgements

This work is supported in part by National Natural Science Foundation of China under grant #60573188.

7. References

[1] Dekai Wu." A Polynomial-Time Algorithm for Statistical Machine Translation," in *Proceedings of ACL 1996*.

[2] Franz Josef Och." Minimum error rate training in statistical machine translation."In *Proceedings of ACL 2003*, pages

[3] Deyi Xiong, Qun Liu and Shouxun Lin. "Maximum Entropy Based Phrase Reordering Model for Statistical Machine Translation," *in Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (COLING/ACL 2006)*. 2006, Sydney, Australia.

[4] Philipp Koehn, Franz J. Och, and Daniel Marcu. "Statistical phrase-based translation," *in Proceedings of HLT-NAACL 2003*, pp. 127-133.

[5] Philipp Koehn. "Pharaoh: a beam search decoder for phrase-based statistical machine translation models," *in Proceedings of the Sixth Conference of the Association for Machine Translation*, America, pp. 115-124.

[6] Franz Josef Och and Hermann Ney. "Discriminative training and maximum entropy models for statistical machine translation," *in Proceedings of the 40th Annual Meeting of the ACL*, 2002, pp. 295-302.

[7] Yang Liu, Qun Liu and Shouxun Lin. "Tree-to-string alignment template for statistical machine translation," *in Proceedings of the 44th Annual Meeting of the ACL*, 2006, Syndney, Australia.

[8] Deyi Xiong, Shuanglong Li, Qun Liu, Shouxun Lin, and Yueliang Qian. "Parsing the Penn Chinese Treebank with semantic knowledge," *in Proceedings of IJCNLP 2005*, pages 70-81.

[9] Franz Josef Och and Hermann Ney. "Improved statistical alignment models," *in Proceedings of the 38th Annual Meeting of the ACL*, 2000, pp. 440-447.

[10] Andreas Stolcke."SRILM – An extensible language modeling toolkit," *in Proceedings of the International Conference on Spoken language Processing*, vol. 2, pp. 901-904