ZZX_MT: the BeiHang MT System for NTCIR-9 PatentMT Task

WenHan Chao
School of Computer Science and Engineering, BeiHang University
37# Xueyuan Rd, Haidian District, Beijing, China, 100191
chaowenhan@buaa.edu.cn

Zhoujun Li
School of Computer Science and Engineering, BeiHang University
37# Xueyuan Rd, Haidian District, Beijing,China, 100191
lizj@buaa.edu.cn

ABSTRACT
In this paper, we describe ZZX_MT machine translation system for the NTCIR-9 Patent Machine Translation Task(PatentMT). We participated in the Chinese-English translation subtask and submit three results, which correspond to three different models or decoding algorithms respectively. Both of the first two are phrase-based SMT approaches integrating the BTG constraint into reordering models, and the last one is a hybrid system, which is an SMT system while using an example-based decoder.

Categories and Subject Descriptors
I.2.7 [**Computing Methodologies**]: ARTIFICIAL INTELLIGENCE – *Natural Language Processing*

General Terms
Experimentation, Languages

Keywords
Phrase-based SMT, BTG, Hybrid MT System, Example-based Decoder.

Team Name: [BUAA]

SubTasks/Languages: [Chinese-English]

External Resources Used: [Giza++, Mmoses, SRILM, Stanford Parser, ICTCLAS]

1. INTRODUCTION
Since the syntax is different between the source and target languages, reordering is necessary in the phase-based Statistical Machine Translation (SMT) systems[1-6]. In the previous stage, researchers provided some local reordering models[1,7-9], most of which only use simple heuristic rules about the positions to limit the reordering. However, they can not explain the complex structures relationship between the source and target language sentences. Thus, for two languages which are very different in syntactic structure, such as Chinese and English, these local models are not enough.

In order to solve the problem, many researchers[10-14] try to introduce the syntactic knowledge into the SMT to constrain the phrase reordering globally, and they have proposed various different ways to make use of syntactic knowledge in SMT. According to Chiang[13], the syntactic knowledge used in the SMT can be divided into two types, one is the linguistic syntax and the other is the formal syntax.

In this paper, we describe our machine translation system ZZX_MT, which is a hybrid system, i.e., it is a log-linear phrase-based statistical machine translation system (PBSMT), while using different sub-models as features, and decoding algorithms, especially it has an example-based decoder.

As a PBSMT, ZZX_MT uses a Tree-Tree model to combine the global and local reordering together. In this model, we will use the BTG for the global reordering, and assume the structure of the BTG tree is independent of the local reordering of the two adjacent phrase pairs, so we can design the global and local reordering models respectively. Also, we provide a tree isomorphism model to use the source-side parser tree to constrain the BTG tree, and provide a local reordering model. Through the tree isomorphism model, we can incorporate the linguistic syntactic knowledge into the formally syntax-based translation model.

In order to consider the global reordering further, ZZX_MT provides an example-based decoder, using the translation example to constrain the translation sentence's structure.

The rest of this paper is organized as follows: section 2 presents ZZX_MT's components; especially tree-isomorphism model and the example-based decoder; in Section 3, we describe the experiments in the PatentMT sub-task[15]. Then, we conclude in Section 4.

2. ZZX_MT System
ZZX_MT system is a modular MT engine, which mainly consists of the following components:

- *Word Alignment:* taking in the bilingual word-aligned training corpus through Giza++, obtains the Viterbi word alignment for each sentence pair, in our system, the word alignment must satisfy the BTG constraint.
- *Model Training:* taking in the bilingual word-aligned training corpus, extracts the valid phrase pairs and builds the **translation model** and the **reordering model**.
- *Decoding*: given a source sentence, search the best translation using the word-aligned corpus and the translation model, reordering model and language model.

2.1 Word Alignment

Word alignment is the basis of the SMT system. In our system, word alignment needs to satisfy the BTG constraint, which is derived from the BTG grammar.

BTG[10] is a synchronous context-free grammar, which generates two output streams simultaneously. It consists of the following five types of rules:

$$A \longrightarrow [AA] \mid < AA > \mid c/e \mid c/\varepsilon \mid \varepsilon/e \qquad (1)$$

where A is the only non-terminal symbol, [] and <> represent the two operations which generate outputs in straight and inverted orientation respectively. c and e are terminal symbols, which represent the phrases in both languages, and ε represents the null word. Each rule is assigned a probability. The first two rules are called combining rules and the last three ones lexical rules.

Since the BTG model only needs to preserve the constituent structure, i.e. the binary branching tree structure, which introduces a weak cross constraint[10], the model achieves a great flexibility to interpret almost arbitrary reordering during the decoding, while keeping a weak reordering constraint in the global scope.

In order to incorporate the BTG information in our translation model, we obtain the word alignment satisfying the BTG constrain.

Thus, after receiving the initial word-aligned bilingual corpus by Giza++, ZZX_MT will run a post-processing word alignment procedure, which uses log-linear word alignment model, which consists of the following features:

- *Conditional Probability Model:* using the p(c|e) and p(e|c) as the base features, which are generated by Giza++.
- *BTG constraint*: counts the number of links in the word alignment, which violating the ITG constraint. In order to ensure that the result word alignment satisfies the constituent structure, we set a very small negative weight for this feature, so that the word alignment will not be used whenever this feature occurs.

In this translation task, since having no the word-alignment develop set, so we did not tune the features' weights and set the weights of features p(c|e) and p(e|c) as 0.5 respectively, and set the weight of BTG constraint as -1000.

2.2 Model Training

After obtaining the word-aligned corpus which satisfying the BTG constrain, we can train our translation models and reordering models.

ZZX_MT is basically a phrase-based SMT system, in order to incorporate the BTG information into the translation system, ZZX_MT regards the process decoding as a sequence of applications of rules in (1), thus we will obtain a BTG tree in the ending of the decoding, i.e. the source and target sentence pair (C,E) will be a derivation D of the BTG.

Thus, ZZX_MT consists of the following models as features:

Lexical rule models

$$h_1(C,T_C,G) = \log \prod_{i=1}^{K} p(e(b_k) \mid c(T_k))$$

$$h_2(C,T_C,G) = \log \prod_{k=1}^{K} p(c(T_k) \mid e(b_k))$$

$$h_3(C,T_C,G) = \log \prod_{k=1}^{K} p_w(e(b_k) \mid c(T_k))$$

$$h_4(C,T_C,G) = \log \prod_{k=1}^{K} p_w(c(T_k) \mid e(b_k))$$

Where, $e(b_k)$ represents the target phrase e in block b_k (i.e., a phrase pair), and $c(T_k)$ is the source phrase c in sub-tree T_k. And $p(e(b_k) \mid c(T_k))$ represents the probability that T_k generates the block b_k ; while $p_w(e(b_k) \mid c(T_k))$ is the corresponding lexical probability, i.e., the internal mapping probability. Similarly, $p(c(T_k) \mid e(b_k))$ represents the probability that b_k maps to T_k , and $p_w(c(T_k) \mid e(b_k))$ is the lexical probability. The four models above correspond to the normal phrase and lexical translation models in the Phase-based SMT, which can be trained through the word-aligned corpus straightforward.

Language model

$$h_5(C,T_C,G) = \log p_{lm}(E)$$

We train the language model using the SRILM, taking in the n-gram order as 3.

Word-Punishment model

$$h_6(C,T_C,G) = I$$

I is the length of the target sentence, which is the number of the words in the target sentence.

Phrase-Punishment model

$$h_7(C,T_C,G) = K$$

K is the number of lexical rules in the BTG tree.

Local reordering model:

$$h_8(C,T_C,G) = \log \prod_{k} p(o_k \mid b_{kl}, b_{kr})$$

Where o_k represents the orientation between two adjacent blocks in the BTG tree, and the value can be *invert* or *straight*. And $p(o_k \mid b_{kl}, b_{kr})$ is the probability of the given orientation involving the two blocks.

We can train this model as follows:

$$r(o \mid b_k, b_{k+1}) = \frac{count(o, b_k, b_{k+1})}{count(b_k, b_{k+1})}$$

Where $count(o, b_k, b_{k+1})$ is the frequencies that the two adjacent blocks are in straight or inverted orientation respectively.

However, due to the constraints for corpus size and the memory of the computer, it is impossible to collect the reordering of any two blocks; and in general, the larger the block is, the smaller the frequency it occurs, so that the reordering probabilities are not accurate.

So, instead of recording all the reordering of any two blocks, we try to use the child blocks to predicate the reordering of any two adjacent blocks. ZZX_MT uses two adjacent child blocks of the two blocks to calculate the probabilities approximately.

Tree isomorphism model:

$$h_9(C, T_C, G) = \Pr(T_G \mid T_C)$$

Where T_C is the parser tree of the source sentence, and T_G is the BTG tree, $\Pr(T_G \mid T_C)$ represents the probability of generating the BTG tree after given the parser tree.

In order to incorporate the syntactic knowledge, we take the syntactic parser tree of the source sentence as the basic framework to generate the BTG tree, trying to make the two trees keep similar; and the BTG tree may tune the structure according to the concrete situations. So the model would be more robust to the syntactic parser tree. The tree-tree model incorporates the linguistically syntactic knowledge, and through the use of the BTG model, it can explain the big difference between the languages.

Besides, when keeping the basic syntactic structure, the model will select the combining orientation based on the local reordering model. Thus, the tree-tree model can reorder the phrases globally based on the trees, and tune the orientations locally.

Additionally, our tree-tree model need not extract translation rules, making it very easy to train the model.

In the following sections, we will describe how to train or calculate the tree isomorphism model.

2.2.1 Tree isomorphism model

Our tree-tree model makes use of the syntactic parser tree of the source sentence to motivate the building of the BTG tree. And the tree isomorphism model ensures the structures of the two trees are similar, and it can also make some revise to increase the explaining ability of the tree-tree model and to strengthen the fault-tolerance to the parser tree.

In this tree-tree model, the so-called isomorphism between the

$$h_9(C, T_C, G) = \sum_{T_k} \delta(T_k \text{cannot find the mapping})$$

parser tree and the BTG tree, refers to that for each sub-tree T_k in the parser tree, we can find a mapping G_{T_k} in the BTG tree, and G_{T_k} is independent, i.e., it is a sub-tree in the BTG tree.

Fig. 1 demonstrates an isomorphic example, where the dot lines represent the successful mapping. Because the "最近 的" and "在哪里" have formed an integration, they can be taken as leaf nodes, so we need only consider the corresponding two sub-trees.

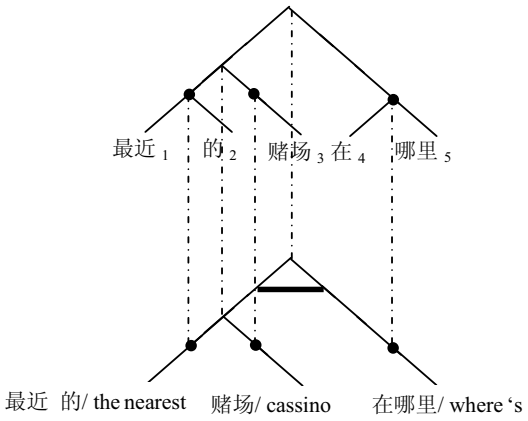

Fig. 1. An example demonstrates the tree isomorphism. The dot lines represent the mapping between sub-trees.

Considering the difference between languages, it must not be fully isomorphic for the BTG tree and the parser tree. So, we can model the isomorphism of them by calculating the similarity between them.

About the tree similarity measure, the straightforward method is to calculate the edit distance, i.e., the operation number to obtain another tree by adding, deleting and modifying nodes in one tree. However, the calculation of the tree edit distance is very complex.

Thus, we propose a simple similarity metric: counting the number of the same valid phrases to calculate the similarity. The underline assumption is: the valid source phrase generated in the parser tree should be likely to be translated as a whole to the target phrase. Then the source phrase and the target phrase will form an independent sub-tree in the BTG tree. So we can use the number of the sub-trees in the BTG tree, which are formed through the valid phrases in the parser tree, to measure the similarity between the two trees.

We call all the phrases that can be extracted in parser tree and BTG tree as "valid phrases", the method to extract them is: extracting the corresponding continuous word sequence of the nodes. And the similarity metric as follows:

$$Sim(T_1, T_2) = \frac{|\, Phrase(T_1) \cap Phrase(T_2) \,|}{|\, Phrase(T_1) \,|}$$

Where T_1 is the parser tree, and T_2 is the corresponding binary branching tree of the source language in the BTG tree. $Phrase(\bullet)$ represents the valid phrases in the tree, and $|\, Phrase(\bullet) \,|$ denotes the number of the valid phrases in the tree.

2.3 Decoding

ZZX_MT system includes two decoders, the first one is a CKY-style decoder, which regard the process of the decoding as a sequence of applications of rules in (1), and taking the above models as features. In order to evaluate the tree isomorphism model, we compare with two results with or without using this model.

2.3.1 The example-based decoder

The example-based [16] decoder consists of two components:

- *Retrieval of examples:* given the input Chinese sentence *C0* and the bilingual word-aligned corpus, collects a set of translation examples {(*C1*, *E1*, *TA1*) ,((*C2*, *E2* , *TA2*),....} from the corpus, where the Ck in each translation example is similar to the input sentence.
- *Decoding*: given the input and the translation examples and the translation models, language models and reordering model, searches the best translation for the input.

During the decoding, we execute the following two steps:

- *Matching*

For each translation example (*Ck*, *Ek*, *TAk*) consists of the BTG tree, we can match the input sentence with the tree, and get some translation templates for each translation example, in which some input words (monolingual phrases) are translated and they must maintain the constituent structure, and some phrases are un-translated. I.e., the template is a partial translation. We call the un-translated phrases as child inputs, and try to translate them using the basic CKY-style decoder.

- *Merging*

If one child input is translated wholly, i.e. no phrase is un-translated. Then, it should be merged into the parent translation template to form a new template. If all child inputs are translated, then returning the final translation. When merging, we must satisfy the BTG constraint.

When decoding, we need to evaluate the translate template using the following function:

$$f(temp) = \log P(E_{trans} \mid C_{trans}) + \log H(C_{untrans})$$

Where $P(E_{trans} \mid C_{trans})$ is the probability for the translated phrases, which can be calculated using the SMT model, and the $H(C_{untrans})$ is the estimated score for the *untranslated* phrases which can also be estimated using the SMT.

3. Experiments

We carried out experiments on the NTCIR-9 Patent Machine Translation subtask: C-to-E, which provides a sentence-aligned training corpus consisting of about 1,000,000 Chinese-English sentence pairs from bilingual patent documents.

We take the following steps to train our models:

1. Preprocessing: segmenting the Chinese sentences using the ICTCLAS; tokenizing the English sentences;
2. Word Alignment: using the Giza++ to train the initial word alignment;
3. Alignment Post-Processing: using our word alignmenter to process the results from step 2, and obtain the word-aligned corpus satisfying the BTG constrain;
4. Model Training: extracting the phrase-pairs and recording the orientation between each phrase-pair with its adjacent phrase-pairs; then calculating the translation model and reordering model.
5. Language Model Training: using the SRILM to train the language model, and the corpus is patent_alt_us2005b only. And the n-gram order is 3.

After trained the models, we tuned the log-linear model using the above features. In order to evaluate the tree model, we turned two models with and without the tree model.

After tuning, we tested our system using the two models and then tested it using the example-based decoder with the tree model.

During the turning and decoding, we used the Stanford parser to parsing the Chinese sentence.

The final results are showed in Table 1 and Table 2 from the committee.

Table 1. Test Results Using the Automatic Metrics

Systems	BLEU	NIST
PBSMT_without_treemodel	0.2631	7.4774
PBSMT_with_treemodel	0.2619	7.4706
ExSMT_with_treemodel	0.2649	7.4924

Table 2. Test results via Evaluated Manually

System	average adequacy	5	4	3	2	1
ExSMT_with_treemodel	3.297	34	76	140	45	5

(a) adequacy

System	pairwise comparison score	tie	AA	A	B	C	F
ExSMT_with_tree model	0.486	0.257	9	23	63	60	145

(b) acceptability

The results showed that the system with example-based decoder will gain improvement than the other two systems. In addition, after analyzing the results, we found that if the system could find the similar examples, it would generate better translated text. However, for we used a strict condition to retrieve the similar examples, most of the source sentences could not find the similar examples. And we will search more effective algorithms to improve this.

However, to our surprise, we found that the system with tree model did not gain better result than the one without tree model. And we need to analysis it further.

4. Conclusions

In this paper, we proposed SMT system with an example-based decoder, which is derived from the BTG, for the NTCIR-9 Patent Machine Translation Task. This approach will take advantage of

the constituent tree within the translation examples to constrain the flexible word re-ordering, and it will also make the omitted words have the chance to be translated. Combining with the reordering model and the translation models in the SMT, the example-based decoder obtains an improvement over the baseline phrase-based SMT system.

5. ACKNOWLEDGMENTS

This research was supported by National Natural Science Foundation of China, Contract No. 61003111; and supported by Research Fund for the Doctoral Program of Higher Education of China (New teacher Fund), Contract No. 20101102120016.

6. REFERENCES

[1] Peter F. Brown, Stepen A. Della Pietra, Vincent J. Della Pietra, and Rorbet. L.Mercer. The Mathematics of Statistical Machine Translation: Parameter estimation. *Computational Linguistics*, 1993, 19(2):263–312.

[2] R. Zens, F. J. Och, and H. Ney. Phrase-based statistical machine translation. In 25th German Conference on Artificial Intelligence (KI2002), Aachen, Germany, September, 2002, pages 18–32.

[3] Franz Joseph 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.

[4] D. Marcu and W. Wong. A phrase-based, joint probability model for statistical machine translation. In Proc. Conf. on Empirical Methods for Natural Language Processing, Philadelphia, PA, July. 2002, pp. 133–139.

[5] Philipp Koehn, Franz Josef Och, & Daniel Marcu. Statistical phrase-based translation. In HLT-NAACL 2003: conference combining Human Language Technology conference series and the North American Chapter of the Association for Computational Linguistics conference series, Edmonton, Canada; May 27- June 1, 2003, pp.48-54.

[6] Richard Zens & Hermann Ney: Improvements in phrase-based statistical machine translation. In HLT-NAACL 2004: Human Language Technology conference and North American Chapter of the Association for Computational Linguistics annual meeting, The Park Plaza Hotel, Boston, USA, May 2-7, 2004, pp. 257-264.

[7] Christoph Tillmann and Tong Zhang. A Localized Prediction Model for Statistical Machine Translation. In Proceedings of the 43rd Annual Meeting of the ACL, 2005, pp. 557-564.

[8] 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 ACL, 2006, pages521-528.

[9] Shanka Kumar, Willianm Byrne. Local Phrase Reordering Models for Statistical Machine Translation. In Proceedings of Human Language Technology Conference and Conference on Empirial Methods in Natural Language Processing (HLT/EMNLP), 2005, pp. 161-168.

[10] Dekai Wu. Stochastic Inversion Transduction Grammars and Bilingual Parsing of Parallel Corpora. *Computational Linguistics*, 1997, 23(3):374.

[11] Kenji Yamada and Kevin Knight. A Syntax-based Statistical Translation Model. In Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics, 2001, pp. 523–530.

[12] Kenji Yamada & Kevin Knight. A decoder for syntax-based statistical MT. In ACL-2002: 40th Annual meeting of the Association for Computational Linguistics, Philadelphia, July 2002, pp.303-310.

[13] David Chiang: A Hierarchical Phrase-Based Model for Statistical Machine Translation. In Proc. of ACL 2005, pages 263–270.

[14] Yang Liu, Qun Liu, & Shouxun Lin. Tree-to-string alignment template for statistical machine translation. In Coling-ACL 2006: Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, 2006, pp.609-616.

[15] 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.

[16] Wen-Han Chao & Zhou-Jun Li: NUDT machine translation system for IWSLT2007. IWSLT 2007: International Workshop on Spoken Language Translation, 15-16 October 2007, Trento, Italy.