The ICT Statistical Machine Translation System for IWSLT 2010

Hao Xiong, Jun Xie, Hui Yu, Kai Liu, Wei Luo,
Haitao Mi, Yang Liu, Yajuan Lii, 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
{xionghao, xiejun, yuhui, liukai, luowei, htmi, yliu, lvyajuan, liuqun}@ict.ac.cn

Abstract

This paper illustrates the ICT Statistical Machine Translation system used in the evaluation campaign of the International Workshop on Spoken Language Translation 2010. We participate in the DIALOG tasks for Chinese-to-English and English-to-Chinese translation respectively. For both tasks, our system has achieved significant improvement with several effective methods as follows: 1) refining the data preprocessing, including Chinese word segmentation, named entity recognition, etc. 2) reducing the number of Out-of-Vocabulary(OOV) on the final test set by applying a fuzzy matching strategy. 3) considering generating a better input for the decoder from the N-best lists of ASR output as a special kind of translation task for the ASR task. 4) improving the performance of every single decoder, and reranking the n-best list for the final results submitted.

1. Introduction

For this year's campaign, we used the following four SMT systems:

- **SuperSilenus**, a linguistically syntax-based system that converts source-forest into target-string with tree-to-string rules acquired from packed forests;
- **TemBruin**, a formally syntax-based system that implements the maximum entropy based reordering model on BTG rules and incorporates some manually written translation templates;
- **John**, a joint tokenization and translation system based on the hierarchical phrase-based model;
- **Moses**, a phrase-based open source system [1].

and participated in two tasks:

- Dialog task, Chinese-English direction;
- Dialog task, English-Chinese direction.

[1]http://www.statmt.org/moses/

We run *John, TemBruin* for Chinese-English task, and *SuperSilenus, Moses* for English-Chinese task respectively. Then rescore the nbest results generated by each system respectively, and this will lead to two rescoring results for each task. Our final submission is chosen from one of these two rescored results which gains the best BLEU score on the development set.

The rest of this paper is organized as follows: Section 2 gives an overview of our four single SMT systems, Section 3 describes the fuzzy matching approach. The rescoring model is depicted in Section 4. Regard generating a better input from the N-best lists of ASR output as a special kind of translation for the ASR task, and this will be presented in details in Section 5. In Section 6, we will report the experiments and results. Finally, section 7 gives a brief conclusion.

2. Systems Overview

2.1. SuperSilenus

SuperSilenus [1, 2] is a linguistically syntax-based SMT system, which employs packed forests in both training and decoding rather than *single-best* trees used in the conventional tree-to-string model [3, 4].

Different from last year's campaign [5], this time we adopt some strategies to improve the performance of this single system. First, we re-implement the fast translation rule matching algorithm [6] to raise the decoding efficiency. Readers can refer to [6] for detailed description. Second, we apply a fuzzy rule matching algorithm based on the intuition that the more rules the decoder could use for translation, the better result it could achieve [7]. The procedure of rule matching in traditional tree-to-string models [3, 4] could be regarded as a string matching process. Our decoder will first traversal the source parse tree, and then try to find the matched translation rules whose source-tree can match the current sub-tree to be concerned. Figure 1 gives an example where the source-tree of the translation rule and the matched sub-tree can both be presented by the string "(IP (NPB) (VP (PP (P (yu) NPB) VPB)))".

However, the previous rule matching approach, namely

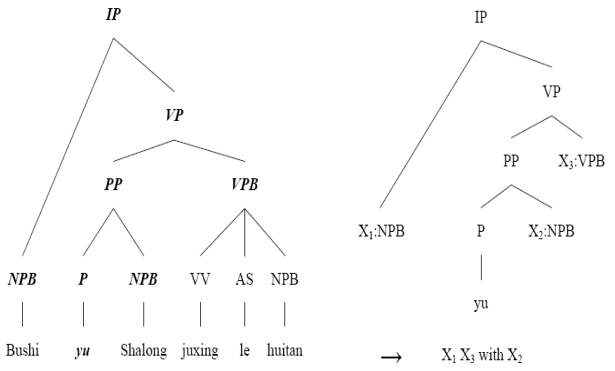

Figure 1: The source parse tree and one matched rule. In the source parse tree, the nodes of matched sub-tree are in bold and italic style.

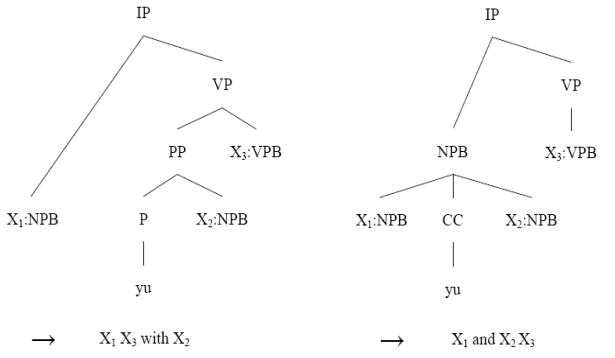

Figure 2: Two translation rules extracted from the training set. Note that the leaf nodes of both source parse trees are the same.

exact matching, implicitly limits the number of available rules. For example, figure 2 shows two different rules extracted from the training set. When performing the exact matching approach, these two rules cannot match a certain sub-tree at the same time. However, this problem can be alleviated when using the following fuzzy matching approach. From figure 2 we find that the alignment information which is truly useful for translation lies only in the leaf nodes for every rule. To employ more rules and weaken the impact of parsing errors, we present a fuzzy matching strategy, in which we only match the root and leaf nodes between the subtrees in source parse tree and the rules. So the two rules in figure 2, which have the same root node "IP" and leaf nodes "NPB", "yu", "NPB" and, "VPB", can be matched simultaneously for a certain subtree. However, following the fuzzy matching approach described above, these two rules may have the same probability for translation, which is not advantageous to the exactly matched one. Therefore, we employ the convolution tree kernel [8] to compute the similarity between the parse tree and the source-tree of the rule, and take the similarity as an extra feature for the decoder.

2.2. TemBruin

TemBruin is a formally syntax-based SMT system, which implements the maximum entropy based reordering model on BTG rules[9], and incorporates manual translation templates in decoding. Based on last year's system, we employ some skills to improve the translation results in this year's campaign.

Typically, there are lots of expressions with relatively fixed pattern in spoken language. It would be beneficial for SMT systems if these pattern are incorporated. Based on this intuition, we have written several translation patterns manually only according to the language phenomenon occurring in training set.

During the decoding phase, these manual translation patterns are utilized in the following manner: first match the input sequence with the source side of every translation pattern. If there are matched patterns in a certain span, additional hypotheses associated with these patterns are generated for this span. Note that these additional hypotheses will coexist with those hypotheses produced by applying traditional BTG rules.

Take the Chinese sentence "这不是息票是登记收据。" as an example. If there is a manual translation pattern "这不是##1是##2。 → this is only ##2 , not ##1 .", which has two variables. Through pattern matching, we can find that the example pattern covers the whole sentence. When the decoder is computing the hypotheses for the span covering the whole sentence, besides the hypotheses produced by applying the BTG rules, some extra hypotheses will be added by combining the right-hand-side of the above pattern and the hypotheses for "息票" and "登记收据".

2.3. John

John is implementation of joint tokenization and translation [10] based on the hierarchial phrase-based model, which takes the Chinese character sequence as input and conducts tokenization and translation simultaneously during decoding. The joint decoder works under the discriminative framework, and employs both tokenization features and translation features. In our implementation, the following 16 features are used as described in [10]:

- 8 traditional translation features as [11]: 4 rule scores (direct and inverse tranlation scores; direct and inverse lexical translation scores); language models of the target side; and 3 penalties for word count, extracted rule and glue rule respectively.
- 8 tokenization features: maximum entropy model, language model and word count of the source side.

Formally, the probability of a derivation D can be represented as

$$P(D) \propto \prod_i \phi_i(D)^{\lambda_i} \quad (1)$$

where ϕ_i are the features mentioned above defined on derivations, and λ_i are feature weights.

2.4. Moses

Moses [12] is a phrase-based model. It is an open source system [2] and uses beam-search to reduce the searching space. We use the default setting for this model in this year's evaluation.

3. Fuzzy Matching

For Chinese-English task, by analyzing the corpus provided by the organizer, we find there exist some words with different spelling styles, but these words do refer to the same thing. Such as "拉斯韦加斯" occurring in the training set and "拉斯维加斯" occurring in the test set, they both mean "Las Vegas". Although there are many "拉斯韦加斯" in the training set, the decoder can not translate "拉斯维加斯" in the traditional manner. To alleviate this problem caused by different spelling style, we present a fuzzy matching approach.

In order to perform fuzzy matching, we need to compute the similarity between different words. Several approaches can meet with this need, but here we choose the string kernel [13]. Formally, the similarity between different strings s and t is $K(s,t)$, where $K(s,t)$ is a kernel function. And we normalize the final score using the following formula.

$$K(s,t) = \frac{K(s,t)}{\sqrt{K(s,s)K(t,t)}} \quad (2)$$

For the details of string kernel function, readers can refer to [13].

The process of fuzzy matching is as follows: First, construct word pairs. The word pair consists of two words, one from the training set, and the other from the test set. Second, compute the similarity for each word pair. Finally, expand the bilingual rule table through substitution of similar words. The similar words for the given lexical word are defined as those words who get the similarity more than 0.8 by performing string kernel method. Additionally, in our experiment, we only consider the words consisting of more than three characters when choosing the similar words.

Note that the fuzzy matching approach mentioned here differs from the fuzzy rule matching approach described in the Section 2.1.

4. Rescoring Model

Our rescoring model is inspired by [14]. We apply the following features.

- 8-gram target language model.
- the ratio between the length of source sentence and that of target sentence.
- the probability of the ratio between the length of source sentence and that of target sentence, which can be predicated through the fitting curve computed on the training set.
- the probability of collocation between two target words, which can be computed on the training set.
- question feature, which indicates whether a sentence is a question sentence.
- the posterior probabilities of the sentence length [15].
- lexicalized reordering rule [16].

Weights of these features are tuned by the MERT tool in Moses package.

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

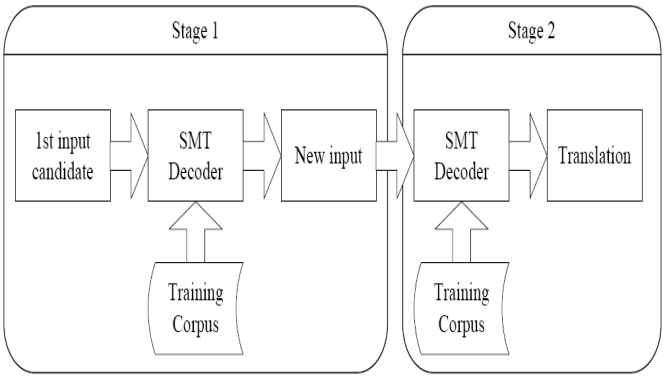

Figure 3: The steps of ASR Translation model

5. ASR Translation Model

According to the introduction of evaluation campaign, the number of N-best list hypotheses of ASR output is not more than 20. In last year's campaign, our method was that the decoder took the whole N-best list hypotheses as input directly, and translated them respectively. Then a rescoring system would select the best result. However, there exist speech recognition errors with incorrect Chinese characters or incorrect English words in almost every input candidate. So it is not reasonable to take the candidates as the decoder's input directly. Therefore, in this year's campaign, we present a novel method to generate a new input from N-best list hypotheses. Our experiment shows that this novel method achieves a better BLEU score than the former method when taking the input of the CRR task as the reference.

Our ASR translation model can be viewed as a two-step translation procedure as shown in Figure 3.

- First, generate the new input from the given N-best list;
- Second, translate the generated input.

The second step is the common translation task, so it would not be depicted in details here. It is reasonable to

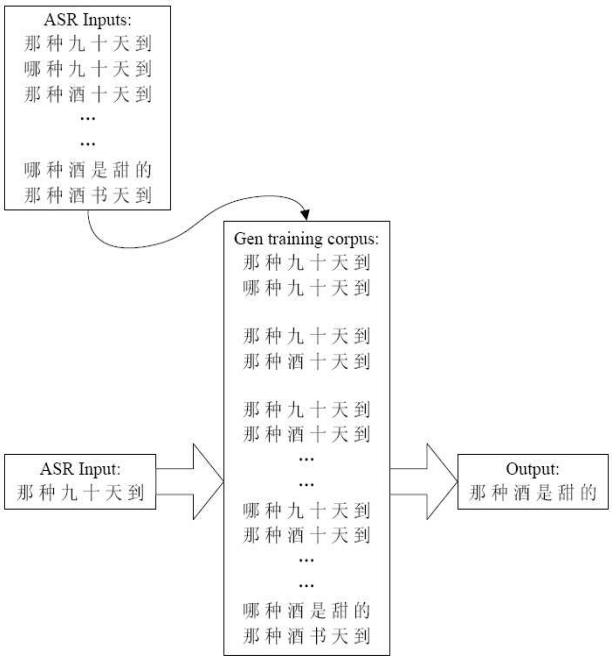

Figure 4: An example of generating new ASR input.

consider the first step as another translation task, a source-to-source translation to some extent. This idea is simple and efficient, because we can apply many decoders to do the translation task. Since it does not require reordering in the problem itself, we choose Moses package in our experiment. However, there does not exist the proper training corpus for Moses to solve the problem. We need to construct the training corpus manually, and our method is as follows: for every two of N-best list hypotheses of ASR output labeled the same sentence id, we generate a sentence pair. Since some characters/words in each ASR output might be correct and the translation of them should be the same as themselves, we generate a special sentence pair, in which the source and the target sentence are the same. So it is easy to see that $k \times k$ sentence pairs are generated for each sentence id.

Figure 4 shows an example of how to generate a new input from the given ASR inputs. The system would select the first ASR original input "那 种 九 十 天 到" as the input, and produce the result "那 种 酒 是 甜 的" as the output of the first step, which is better than any original ASR input.

For tuning the feature weights, we construct the development set by using the ASR sentences as source, and the CRR input as reference. With these manual set, we could run Moses package to produce the new input for the second step.

6. Experiments

6.1. Data Preparation

In this year's evaluation, we only use the data provided by the organizer.

6.1.1. Chinese Segmentation

Different Chinese segmentation tools would give different segmentation, leaving how to incorporate these segmentation into a final one to be an important strategy which we need to choose carefully. According to the feature of the original and ICTCLAS3.0[3] segmentation, we develop three strategies to modify the training set and test them in the experiments.

- refine the original segmentation by the ICTCLAS segmentation(short for *ORI++*).
- remove the space of the original segmentation and re-segment by the open toolkit ICTCLAS(short for *ICTCLAS*).
- reserve both the original segmentation and the ICTCLAS segmentation, and combine them in the training set(short for *COMB*).

We find that the granularity of the original segmentation is relative coarser than that of the ICTCLAS segmentation, and the named entity is typically not segmented. For example, "城市酒店" is kept in the original segmentation, while ICTCLAS will segment it into "城 市" and "酒 店". Because coarse-grained segmentation will lead to data spareness, especially on the relative small corpus. To alleviate the spareness problem, for strategy *ORI++*, we briefly replace the original segment which has more than four Chinese characters with the corresponding ICTCLAS segmentation result. Furthermore, we also try to combine the temporal and numerical expressions using heuristic rules, for both ICTCLAS and original segmentation.

Table 1: *The experimental results of different segmentation strategies on dev set.*

	ORI++	ICTCLAS	COMB
c2e	**54.33**	53.80	53.59
e2c	**47.62**	46.92	46.28

Table 1 shows the experimental results of three segmentation strategies mentioned above. We can find that the strategy *ORI++* achieves the best results on the dev set. The result is not surprising, because the original segmentation is relatively good and refined by ICTCLAS segmentation with heuristic rules. And the reason why the strategy *COMB* gets the lowest score may exist in that it reduces the quality of the word alignment. For example, when combining the two segmentation results, both the probability of "城市酒店" aligned to "City Hotel" and that of "城市 酒店" aligned to "City Hotel" will be lower than that in the other two strategies.

[3]http://www.nlp.org.cn

6.1.2. English Lowercase and Tokenization

In order to weaken the problem of data spareness emerging from the relative small training corpus, we make all English letters lowercase, and apply a rule-based tokenizer realized by ourselves to the corpus.

6.1.3. Alignment

We run GIZA++ and use the "grow-diag-final" heuristic to get the many-to-many word alignment. However, we find that the alignment generated by GIZA++ tends to have high recall but low precision. So we perform comparative experiments using Berkeley Aligner [4] which tends to high precision but low recall.

Table 2: *The experimental results of different alignment strategies on dev set.*

	GIZA++	Berkeley	GIZA++&Berkeley
CE	54.03	52.78	**54.33**
EC	45.09	43.97	**47.62**

Table 2 presents the results of different alignment strategies on the development set. And we can find that the strategy of combining two alignments achieves the best BLEU score, because both the precision and recall are higher than the single alignment.

6.1.4. Others

We use the SRI Language Modeling Toolkit [17] to train the Chinese/English 5-gram language model with Kneser-Ney smoothing on the Chinese/English side of the training corpus respectively.

Regarding to SuperSilenus, we apply the Chinese parser of [18] and English parser of [19] to parse the source and target side of the bilingual corpus into packed forests respectively. Then we prune the forests with the marginal probability-based inside-outside algorithm [20] with a pruning threshold $p_e = 3$. At the decoding phase, we use a large pruning threshold $p_d = 12$ to generate the packed forest.

As mentioned in Section 2.1, we employ a fuzzy rule matching approach for SuperSilenus. And table 3 presents the translation results on development set. We can see that the fuzzy rule matching approach improves the performance of tree-based system significantly and also make the BLEU score increase by 0.7 in forest-based system.

As mentioned in 2.2, we have written some manual translation patterns on the basis of the training set for TemBruin. Table 4 shows the overall number of patterns and the times of these patterns matching sentences on the development set. And table 5 presents the count of matching sentences on the test set. From table 5 we can see that the amount of matching sentences on English-Chinese task is limited. The main reason is that we have not written many patterns for English-Chinese task.

Additionally, we use the overall supplied development corpus as our development set to tune feature weights.

[4] http://nlp.cs.berkeley.edu/Main.html#WordAligner

Table 3: *The experimental results of different tree-to-string models on dev set. (t2s is tree based tree-to-string translation model and f2s is forest based tree-to-string translation model)*

System	Dev(BLEU 4)
t2s(exactly matching)	45.32
t2s(fuzzy matching)	**46.54**
f2s(exactly matching)	46.98
f2s(fuzzy matching)	**47.62**

Table 4: *The amount of patterns and statistics on dev set.*

	Pattern Number	Matching Counts on Dev Set
CE	548	670
EC	88	200

6.2. Experimental results

According to the results on development set, we submit the rescored result of John for Chinese-English translation and that of SuperSilenus for English-Chinese translation, respectively. Table 6 gives the final scores of each system on test set.

From table 6 we can find that the reranking technique achieves improvement over single system. Importantly, we can also find that John gains better results than the other one on Chinese-English translation. The main reason lies in that John is a joint tokenization and translation system which generates more acceptable Chinese word segmentation for translation and alleviate the propagation of segmentation error.

Table 6 also shows that SuperSilenus achieves better results in English-Chinese translation, compared with the performance in Chinese-English translation. Since SuperSilenus is a forest based tree-to-string translation model, the precision of parsing will affect the performance of translation. In contrast with parsing Chinese sentences, it is more acceptable when parsing English sentences.

We also contrast the results of this year with those of last year's on the 09 progress test set. From the table 7, it is clear to see that our system achieves significant improvement over last year's. The following are the reasons that can account for this: 1) the training corpus of this year is larger than that of last year. 2) most work has been refined to produce better

Table 5: *Statistics of matching sentences on test set.*

Input	Matching Counts
09CE.CRR	154
09CE.ASR	147
10CE.CRR	121
10CE.ASR	130
09EC.CRR	23
09EC.ASR	28
10EC.CRR	41
10EC.ASR	40

Table 6: *The BLEU scores of each system on test set (case insensitive).*

Task	Input	System	BLEU
CE	CRR	Rescoring(John)	**24.58**
		John	23.77
		TemBruin	23.70
CE	ASR.20	Rescoring(John)	22.20
		John	**22.27**
		TemBruin	19.35
EC	CRR	Rescoring(SuperSilenus)	**37.67**
		SuperSilenus	35.16
		Moses	33.44
EC	ASR.20	Rescoring(SuperSilenus)	**30.80**
		SuperSilenus	28.96
		Moses	28.17

output.

7. Conclusions

In this paper, we introduce the ICT statistical machine translation system for the evaluation campaign of IWSLT 2010. First, we have used three single systems for each task and then do the result reranking. Finally we submit the reranking result which achieves the best BLEU score on the development set. For this year's campaign, we have done the following effective work: 1) refine the data preprocessing. 2) adopt a fuzzy matching technique to reduce the number of OOV. 3) apply a novel method for the ASR task. 4) improve the performance of every single decoder, and rerank the n-best list for the final results submitted. These work helps our system achieves significant improvement over last year's.

8. Acknowledgements

The authors were supported by National Science Foundation of China, Contracts 60736014 and 60873167, and 863 State Key Project No. 2006AA10108. We thank two anonymous reviewers for their thoughtful suggestions.

Table 7: *The experimental results of this year and last year on 09 test set (case insensitive).*

Task	Input	System	BLEU
CE	CRR	last year	31.85
		this year	**36.70**
CE	ASR.20	last year	28.53
		this year	**33.34**
EC	CRR	last year	39.98
		this year	**49.61**
EC	ASR.20	last year	29.85
		this year	**38.57**

9. References

[1] H. Mi, L. Huang, and Q. Liu, "Forest-based translation," in *Proceedings of ACL*, 2008.

[2] H. Mi and L. Huang, "Forest-based translation rule extraction," in *Proceedings of EMNLP*, Honolulu, Hawaii, October 2008, pp. 206–214.

[3] Y. Liu, Q. Liu, and S. Lin, "Tree-to-string alignment template for statistical machine translation," in *Proceedings of COLING-ACL*, Sydney, Australia, July 2006, pp. 609–616.

[4] Y. Liu, Y. Huang, Q. Liu, and S. Lin, "Forest-to-string statistical translation rules," in *Proceedings of ACL*, Prague, Czech Republic, June 2007, pp. 704–711.

[5] H. Mi, Y. Liu, T. Xia, X. Xiao, Y. Feng, J. Xie, H. Xiong, Z. Tu, D. Zheng, Y. Lv, and Q. Liu, "The ICT Statistical Machine Translation Systems for the IWSLT 2009," in *Int. Workshop on Spoken Language Translation Evaluation Campaign on Spoken Language Translation*, 2009, pp. 55–59.

[6] H. Zhang, M. Zhang, H. Li, and C. Tan, "Fast translation rule matching for syntax-based statistical machine translation," in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2-Volume 2*. Association for Computational Linguistics, 2009, pp. 1037–1045.

[7] D. Chiang, "Learning to translate with source and target syntax," in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2010, pp. 1443–1452.

[8] M. Collins, P. Koehn, and I. Kucerova, "Clause restructuring for statistical machine translation," in *Proceed-*

ings of ACL. Ann Arbor, Michigan: Association for Computational Linguistics, June 2005, pp. 531–540.

[9] D. Xiong, Q. Liu, and S. Lin, "Maximum entropy based phrase reordering model for statistical machine translation," in *Proceedings of COLING/ACL 2006*, 2065, pp. 521–528.

[10] X. Xiao, Y. Liu, Y. Hwang, Q. Liu, and S. Lin, "Joint tokenization and translation," in *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*. Beijing, China: Coling 2010 Organizing Committee, August 2010, pp. 1200–1208.

[11] D. Chiang, "A hierarchical phrase-based model for statistical machine translation," in *Proceedings of ACL*, Ann Arbor, Michigan, June 2005, pp. 263–270.

[12] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, *et al.*, "Moses: Open source toolkit for statistical machine translation," in *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*. Association for Computational Linguistics, 2007, pp. 177–180.

[13] H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, and C. Watkins, "Text classification using string kernels," *The Journal of Machine Learning Research*, vol. 2, pp. 419–444, 2002.

[14] L. Shen, A. Sarkar, and F. Och, "Discriminative reranking for machine translation," in *Proceedings of HLT-NAACL*, 2004, pp. 177–184.

[15] R. Zens and H. Ney, "N-gram posterior probabilities for statistical machine translation," in *Proceedings of the Workshop on Statistical Machine Translation*. Association for Computational Linguistics, 2006, pp. 72–77.

[16] B. Chen, M. Cettolo, and M. Federico, "Reordering rules for phrase-based statistical machine translation," in *Int. Workshop on Spoken Language Translation Evaluation Campaign on Spoken Language Translation*, 2006, pp. 1–15.

[17] A. Stolcke, "Srilm - an extensible language modeling toolkit," in *Proceedings of ICSLP*, vol. 30, 2002, pp. 901–904.

[18] D. Xiong, S. Li, Q. Liu, and S. Lin, "Parsing the penn chinese treebank with semantic knowledge," in *Proceedings of IJCNLP 2005*, 2005, pp. 70–81.

[19] E. Charniak and M. Johnson, "Coarse-to-fine-grained n-best parsing and discriminative reranking," in *Proceedings of the 43rd ACL*, 2005.

[20] L. Huang, "Forest reranking: Discriminative parsing with non-local features," in *Proceedings of ACL*, 2008.