# NEUTrans: a Phrase-Based SMT System for CWMT2009

Tong Xiao, Rushan Chen, Tianning Li, Muhua Zhu,

Jingbo Zhu, Huizhen Wang and Feiliang Ren

132#, Natural Language Processing Lab
Northeastern University, Shenyang 110004, China
E-mail：{xiaotong, zhujingbo, wanghuizhen, renfeiliang}@mail.neu.edu.cn
{chenrs, litn, zhumh}@ics.neu.edu.cn

**Abstract:** *In this report, we describe our (NEUNLPLab) phrase-based statistical machine translation (SMT) system (NEUTrans) for the participation of news domain Chinese-to-English single-system translation task in the 5th China workshop on Machine Translation (CWMT2009). We submitted four translation results for this task. In this report, we first give an introduction of the framework and the key techniques used in our system, then analyze on the experimental results, and finally discuss the issues we found during the development of the system.*

**Keywords:** *natural language processing, machine translation, phrase-based statistical machine translation*

## 1 Introduction

We (NEUNLPLab) developed a phrase-based SMT system (NEUTrans) and participated in the news domain Chinese-to-English single-system translation task in the 5th China workshop on Machine Translation (CWMT2009[1]). In this report, we describe the framework and the key techniques used in NEUTrans. In addition, we analyze the experimental results over the SSMT07 and CWMT08 evaluation sets, and give a discussion on some issues we found during the development of the system.

## 2 System Description

NEUTrans is basically in a phrase-based SMT framework and can be regarded as an application of bracketing transduction grammar (BTG) in MT (Wu, 1996). Under BTG scheme, all possible reorderings can be compactly represented with binary bracketing constraints. Besides, BTG-based decoding can be easily implemented with the CKY parsing algorithm. For these reasons, we design NEUTrans in the BTG framework. NEUTrans is based on the log-linear model (Och and Ney, 2002), in which a set of features are combined in a log-linear way. The default features are the same as Moses's (Koehn et al., 2007). Figure 1 shows the framework of NEUTrans.

(1) The Chinese sentences are segmented using the Chinese segmentation toolkit developed by NEUNLPLab[2]. The English sentences are tokenized with a rule-based English tokenizer, and the case information is removed. Before training, all the numbers in both source and target-languages are replaced by a symbol in order to alleviate the data sparseness. In decoding, numbers are translated using a rule-based translation sub-system in advance.

(2) Chinese named entities (NE), such as person name (PER), organization names (ORG), location names (LOC), are recognized using a CRF-based NE recognizer. Instead of translating NE with the translation model during decoding, we obtain the translation of NEs with a rule-based NE translation sub-system as well as a bilingual NE dictionary consisting of 10, 000 bilingual NE pairs.

(3) Word alignment is performed on the bilingual sentences with the open source toolkit GIZA++[3]. After obtaining word alignment in both directions, we refine the alignments with the symmetrization method proposed in (Xiao et al, 2009) to get the symmetric word alignment.

[1] http://www.icip.org.cn/cwmt2009/
[2] http://www.nlplab.com/
[3] http://code.google.com/p/giza-pp/

(4) Stanford parser[4] is used to get the dependency parse trees on the target-side for dependency language model training.

(5) Both the traditional *n*-gram language model and dependency language model are trained using the open source language modeling toolkit SRILM[5].

(6) To recover the case information, we use the recaser provided within Moses SMT toolkit[6] which combines a rule-based recasing sub-system and a HMM-based recasing sub-system.

In the rest parts of this section we briefly describe the key techniques used in NEUTrans.

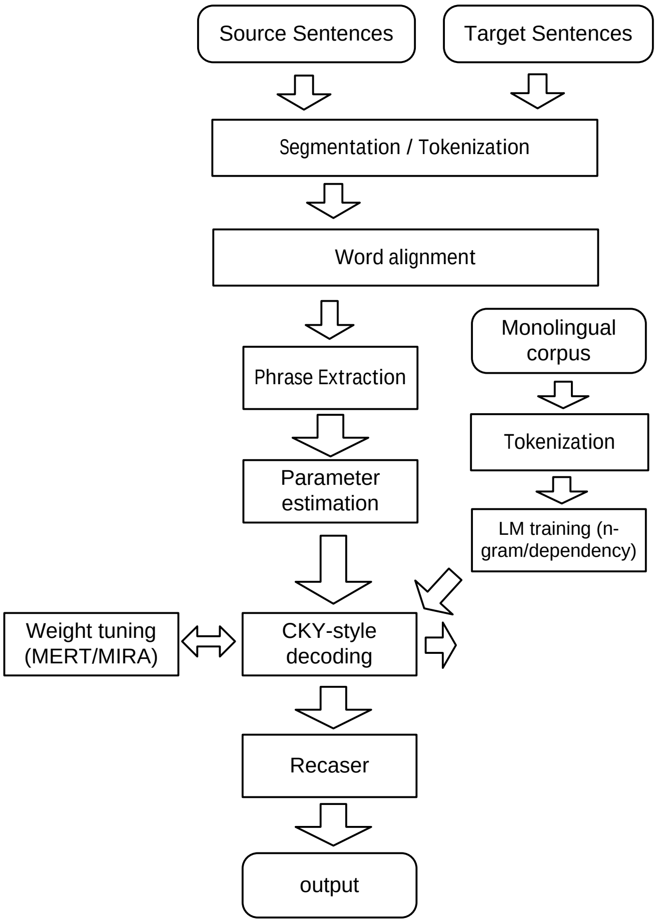

Figure 1. Architecture of NEUTrans

## 2.1 Reordering models

In NEUTrans three reordering (distortion) models are taken together to model the reordering problem.

(1) Calley and Manning (2008)'s hierarchical reordering model, which can be seen as the improved reordering model of Moses's. Differing from (Koehn et al., 2007; Calley and Manning, 2008), our CKY-style decoder guarantees that the translated source words cover contiguous span on the source-side, thus the features used to model the phrase orientations in both source-to-target and target-to-source directions are adopted in our system. Therefore there are totally 12 (6 for source-to-target and 6 for target-to-source) reordering features of this model used in our system.

(2) Maximum Entropy (ME)-based lexicalized reordering model proposed by Xiong et al. (2006). In NEUTrans the maximum entropy model is trained using our implementation of the GIS algorithm.

(3) Structural reordering model proposed in (Chiang et al., 2008). In NEUTrans only the course-grained structural reordering features are used.

[4] http://nlp.stanford.edu/software/lex-parser.shtml

[5] http://www.speech.sri.com/projects/srilm/manpages/

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

### 2.2 Language models

The basic language model is an *n*-gram language model (Jelinek, 1998) which is widely used in the MT field. In NEUTrans, we use multiple *n*-gram languages ranging from bi-gram to 5-gram, which can be regarded as an interpolation of the language models with different orders.

Besides the traditional *n*-gram language model, a target dependency language model is also used to further improve the system. The basic idea is nearly the same as (Shen et al., 2008) which extends a hierarchical SMT system (Chiang, 2005) with the string-to-dependency model. In this model, we 1) extract the so-called *well-formed* phrases with the dependency structures on the target-side; then 2) make use of them to construct the dependency structures for each partial translation; and finally 3) calculate the dependency language model score with respect to the dependency structures generated during decoding.

### 2.3 Training of log-linear model

To estimate the feature weights of the log-linear model, two state-of-the-art methods - Minimum Error-Rate Training (MERT) (Och, 2003) and discriminative training (Chiang et al., 2008), are utilized. Though MERT is simple and effective for the log-linear model training of SMT systems, it does not always work reliably when the number of features exceeds a certain number (e.g. 30). Addressing this problem, using discriminative training method, Margin Infused Relaxed Algorithm (MIRA) is a nice solution (Chiang et al., 2008).

Since NEUTrans has over 40 features, we also use MIRA for log-linear model training in order to get more stable performance. The only difference between our implementation and Chiang et al.'s (2008) lies in that we take the highest-BLEU translation as the oracle translation, while Chiang et al. choose the oracle translation in terms of the combination of BLEU score and model score.

## 3 Experiments

We conduct experiments over the SSMT07[7] and CWMT08[8] Chinese-to-English evaluation sets to test the performance of our system.

### 3.1 Experimental setup

For phrase learning and reordering model training, we use all the bilingual data provided within CWMT09, including over 3.8 million sentence pairs. A tri-gram dependency language model and a 5-gram traditional language model are trained on a corpus of 10 million English sentences containing the English side of the training corpus and Reuters news corpus. The English recaser is trained on the English side of the training corpus. The development set for weight tuning comes from China 863-2005 MT evaluation task which contains 489 Chinese sentences and 4 reference translations for each Chinese sentence. The test sets are the evaluation sets used in SSMT07 (1002 Chinese sentences) and CWMT08 (1006 Chinese sentences).

The default method of weight tuning is MERT. The beam size is set to 30 in all the experiments. Cube pruning (Huang and Chiang, 2007) is applied to reduce the search space. All the results are reported in case-insensitive NIST version BLEU4(%).

### 3.2 Effect of reordering models

Table 1 shows the BLUE scores of our system with different reordering models. The dependency language model is not used in this set of experiment. For the comparison, the performance of Moses is also reported.

We can see that both the hierarchical reordering model and ME-based reordering model outperform the baseline model (Moses). They achieve 0.8/0.5 BLEU point improvement on SSMT07 test set and 0.7/0.7 BLEU point improvement on CWMT08 test set. As expected, the structural reordering model performs worse than the baseline model due to the simple modeling of the reordering problem. To examine the effect of using multiple reordering models, we also use

[7] http://taalunieversum.org/agenda/1012/ssmt_2007/

[8] http://nlpr-web.ia.ac.cn/cwmt-2008/

both of the hierarchical reordering model and the ME-based reordering model in our system. The BLEU scores (Row 6) show that the performance can be further improved by a linear combination of these two models. But when all the reordering models are integrated together, there is no further significant improvement (Row 7). The improvement achieved by the system with multiple reordering models indicates that our system can benefit from each individual reordering model by using them together.

The results in Table 1 verify that the Moses's reordering model can be improved by better training method and the use of source-to-target reordering features. Besides, we think that the improvement made by ME-based reordering probably lies in that word-based features can alleviate the data sparseness caused by phrase-based features.

Table 1. Comparison of different reordering models.

| System | Dev. | Test-SSMT07 | Test-CWMT08 |
|---|---|---|---|
| Moses | 24.8 | 27.3 | 27.3 |
| Basic system + hierarchical reordering | 25.5 | 28.1 | 28.0 |
| Basic system + ME-based reordering | 25.4 | 27.8 | 28.0 |
| Basic system + structural reordering | 23.9 | 26.5 | 26.9 |
| Basic system + hierarchical reordering + ME-based reordering | 25.8 | 28.5 | 28.5 |
| Basic system + all the reordering models | **25.9** | **28.6** | **28.5** |

### 3.3 Effect of multiple LM models

In the second set of experiment, we test the system performance when more than one language models are integrated. Since the system with three reordering models achieves the best performance in the previous experiment, we take it as the baseline system. Table 2 shows the experimental results.

BLEU scores in Row 3 show that the baseline system can be improved by using multiple traditional *n*-gram languages ranging from bi-gram to 5-gram, which can be regarded as an interpolation of the language models with different orders. We also test the system performance when the 5-gram traditional language and the tri-gram dependency language model are used together (Row 4). Unfortunately, the dependency language model does not show great power in the experiment. Instead, it behaves slightly worse than the baseline system on the test set of SSMT07. When all the language models are integrated into our system (Row 5), our system achieves the best performance. But it is just slightly better than “Baseline + bi-gram~4-gram” on the development test and SSMT07 test set.

The results shown in Table 2 indicate that our system can benefit from the interpolated *n*-gram model, but the dependency language model is not helpful enough.

Table 2. Comparison of different language model settings.

| System | Dev. | Test-SSMT07 | Test-CWMT08 |
|---|---|---|---|
| Baseline (5-gram) | 25.9 | 28.6 | 28.5 |
| Baseline + bi-gram~4-gram | 26.3 | 29.2 | **29.0** |
| Baseline + Dep-LM | 26.0 | 28.5 | 28.6 |
| Baseline + bi-gram~4-gram + Dep-LM | **26.4** | **29.3** | **29.0** |

### 3.4 Effect of training method

In previous experiments, the best performance is achieved by using all the features, where the total number of features is over 40. As is discussed in (Chiang et al., 2008), MERT is unreliable when the number of feature weights is larger than a certain number (maybe 30). To alleviate this problem, MIRA is used as a substitute for MERT. Table 3 shows comparison result of MERT and MIRA.

We can see that MIRA outperforms MERT over 0.6 BLEU scores on the development set, while it achieves much lower performance on both test sets. This unpromising result suggests that MIRA probably overfits the development set in the experiment.

Table 3. Comparison of MERT and MIRA

| System | Dev. | Test-SSMT07 | Test-CWMT08 |
|---|---|---|---|
| Baseline (all features) + MERT | 26.4 | **29.3** | **29.0** |
| Baseline (all features) + MIRA | **27.0** | 28.8 | 28.4 |

### 3.5 Final submissions

According to the experimental results, we produce four submissions with different settings. Our "primary" system is based on all of the 3 reordering model and the bi-gram~5-gram interpolated *n*-gram language model. Our "contrast-b" and "contrast-c" systems are nearly the same as our "primary" system. The difference lies in that "contrast-b" uses MIRA instead of MERT as the training method of log-linear model, and "contrast-c" uses the evaluation set of CWMT08 as the development set and enforces the length(output)/length(input) to be 1.085 which is slightly larger than that of the development set. "contrast-d" system is built by adding the target-dependency language model into our "primary" system. The performance of theses four systems is shown in Table 4. It should be noted that the performance on the test set of CMWT09 is reported in case-sensitive BLEU4-SBP(%).

Table 4. Performance of the systems for final submissions

| System | Dev. | Test-SSMT07 | Test-CWMT08 | Test-CWMT09 (BLEU-SBP) |
|---|---|---|---|---|
| Primary | 26.3 | 29.2 | **29.0** | 22.6 |
| Contrast-b | **27.0** | 28.8 | 28.4 | 22.7 |
| Contrast-c | N/A | 28.2 | 28.2(dev) | **23.0** |
| Contrast-d | 26.4 | **29.3** | **29.0** | 22.1 |

# 4 Discussion

We think that the most effective techniques used in our system are the multiple reordering models and the interpolated *n*-gram language model. They yield 1 BLEU point improvement over the baseline system (Moses). But other techniques, such as target dependency language model, are not as effective as we expected. We think that some interesting issues needs to be discussed and addressed in our future work.

(1) In our experiments, dependency language model seems not as effective as described in the related work (Shen et al., 2008). There might be two reasons to explain why dependency language model does not work in our system. First, the coverage of well-formed phrase is low. We find that the over 30% entries of phrase table are the phrases covered with ill-formed structures. These phrases make the dependency structures generated during decoding relatively more unreliable. Second, the quality of training data is unsatisfactory. It is found that a great number of useless sentences in the training data, such as long sequence of numbers extracted from tables. The quality of dependency parse trees is another problem. The second problem can be alleviated by using a better dependency parser and high-quality training data, such as GIGAWORD of LDC. Actually in our recent experiments, using better and larger data for dependency language training can yield about 0.5 BLEU point improvement over our "primary" system. The first problem is an interesting topic, and we intend to study it in the further.

(2) MIRA does not work well in our experiments. As discussed in Section 3.4, overfitting may be the major problem. We think that it can be alleviated by using larger *n*-best lists of candidates in weight tuning, since more candidates explored in training can generally reduce the risk of overfitting.

(3) We manually analyzed the errors in the translations produced by our system, and found that

missing translation is a serious problem (over 10%). We think that there are 2 reasons. 1) Using evil feature of word deletion. The evil feature makes the system tend to generate comparatively shorter translation. Though this tricky feature is effective in most of the automatic-evaluation-oriented MT tasks, it generally results in low translation quality in manual evaluation due to the lost of the translation of content words. Addressing this problem, we intend to model the problem of word deletion with contextual information, e.g. using the context-sensitive models for word deletion (Li et al., 2008). 2) Low coverage of phrase table. Though over 3.8 million sentence pairs are used for training, only 1% entries of the phrase table are matched in our system. The quality of bilingual corpus is somewhat a problem.

(4) Our system works on each individual sentence, while the input text is generally segmented in both document-level and sentence-level. In error analysis, we found that some difficult problems, such as translation of abbreviation, the problem of coreference and omission of subject, can be alleviated if we enlarge the scope of translation from sentences to documents. Document-level consensus-based MBR decoding might be one of the possible solutions. Document translation is an interesting and attractive issue, and we intend to study it in our future work.

## 5 Conclusion

We developed a phrase-based SMT system (NEUTrans) and participated in the news domain Chinese-to-English single-system translation task in the 5th China workshop on Machine Translation . In the future, we intend to share our system and the source code publicly.

## 6 Acknowledgements

This work was supported in part by the National Science Foundation of China (60873091).

## References

David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proc. of* ACL2005, Ann Arbor, Michigan, pages: 263-270.

David Chiang, Yval Marton and Philip Resnik. Online Large-Margin Training of Syntactic and Structural Translation Features, In *Proc. of* EMNLP2008, Honolulu, pages: 224-233.

Michel Galley and Christopher D. Manning. 2008. A Simple and Effective Hierarchical Phrase Reordering Model. In *Proc. of* EMNLP2008, Hawaii, pages: 848-856.

Chi-Ho Li, Dongdong Zhang, Mu Li, Ming Zhou, Hailei Zhang. 2008. An Empirical Study in SourceWord Deletion for Phrase-based Statistical Machine Translation. In *Proc. of* Workshop on Statistical Machine Translation, Honolulu, pages: 735-744.

Liang Huang and David Chiang. 2007. Forest rescoring: Faster decoding with integrated language models. In *Proc. of* ACL 2007, , Prague, Czech Republic, pages: 144-151.

F. Jelinek. 1998. Statistical Methods for Speech Recognition. MIT Press, Cambridge, MA.

Philipp Koehn, Hieu Hoang, Alexandra Birch Mayne, Christopher Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In ACL 2007 demonstration session.

Franz Och and Hermann Ney. 2002. Discriminative Training and Maximum Entropy Models for Statistical Machine Translation. In *Proc. of* ACL2002, Philadelphia, pages: 295-302.

Franz Och. 2003. Minimum Error Rate Training in Statistical Machine Translation. In *Proc. of* ACL2003, Japan, pages: 160-167.

Libin Shen, Jinxi Xu and Ralph Weischedel. 2008. A New String-to-Dependency Machine Translation Algorithm with a Target Dependency Language Model. In *Proc. of* ACL2008, Columbus, Ohio, USA, pages: 577-585.

Dekai Wu. 1996. A polynomial-time algorithm for statistical machine translation. In *Proc. of* ACL1996, California, USA, pages: 152–158.

Tong Xiao, Tianning Li, Rushan Chen, Jingbo Zhu and Huizhen Wang. 2009. Word Re-alignment for Statistical Machine Translation. In *Proc. of* CNCCL2009, China, pages: 439-445. (in Chinese)

Deyi Xiong, Qun Liu and Shouxun Lin. 2006. Maximum Entropy Based Phrase Reordering Model for Statistical Machine Translation. In *Proc. of* ACL 2006, Sydney, pages: 521-528.