The USTC Machine Translation System for IWSLT 2014

Shijin Wang[+*], *Yuguang Wang*[*], *Jianfeng Li*[*], *Yiming Cui*[*], *Lirong Dai*[+]

[+]National Engineering Laboratory of Speech and Language Information Processing,
University of Science and Technology of China
[*]IFLYTEK Co. LTD.
shijinwang@ustc.edu
{ygwang2, jfli3, ymcui}@iflytek.com
lrdai@ustc.edu.cn

Abstract

This paper describes the University of Science and Technology of China's (USTC) system for the MT track of IWSLT2014 Evaluation Campaign. We participated in the Chinese-English and English-Chinese translation tasks. For both tasks, we used a phrase-based statistical machine translation system (SMT) as our baseline. To improve the translation performance, we applied a number of techniques, such as word alignment with the l_0-norm, phrase table smoothing, hierarchical reordering model, domain adaptation of the language and translation model, recurrent neural network based language model, neural network joint model, etc. By integrating these techniques, we obtained total improvements of 4.2% BLEU score for Chinese-English system and 3.7% BLEU score for English-Chinese system, compared to the baseline systems.

1. Introduction

In the IWSLT 2014 evaluation campaign, we participated in the optional MT track with the Chinese-English and English-Chinese translation tasks. We build a phrase-based statistical machine translation system for these tasks, and similar techniques are applied to Chinese-English and English-Chinese systems.

Before training, Chinese sentences are segmented into words using our Chinese word segmentation tool, and English sentences are tokenized and transformed into lower case. After preprocessing, GIZA++ is applied for training word alignments. Then, bilingual phrase pairs are extracted from word aligned parallel sentences. Based on the extracted phrase table, we build a weak baseline system with several widely-used features. The feature weights are tuned using Minimum Error Rate Training (MERT) [1].

By refining some steps in the training process we obtained our strong baseline. Firstly, we tried different development set. Secondly, we modified GIZA++ with the l_0-norm [2]. Then we tried different heuristics to combine bidirectional word alignment results. When calculating the phrase translation probabilities, we adopted Good-Turing smoothing rather than using relative frequency. By also using hierarchical reordering model (HRM) and k-best Margin Infused Relaxed Algorithm (kbMIRA) [3], our strong baseline system obtained significant improvements over the weak baseline.

To further improve translation performance, we exploited additional models, including more and larger language models, neural network based models, out-of-domain models trained from MultiUN corpus, and an operation sequence model [4]. We use these models in two ways: one is to integrate them into the decoder, and the other is to use them to rerank the n-best translations generated by the decoder.

Language models play an important role in our statistical machine translation system. Besides the in-domain language model trained from the TED training corpus, we built several larger language models from English Gigaword corpus and News Crawl corpora provided by the evaluation campaign. These language models were added into the translation system as separate features. We also built a word class based language model to alleviate data sparseness. Furthermore, a backward language model is used in reranking.

Neural networks have been successfully applied to machine translation recently. In our system, we built a recurrent neural network language model (RNNLM) for reranking. We also built several neural network joint models (NNJM), one for decoding, and the others for reranking.

The rest of the paper is organized as follows. In section 2, we generally describe the techniques we adopted in the translation systems. In section 3, we illustrate our experimental results on Chinese-English and English-Chinese translation systems. In the last section, we give a brief conclusion and the future work.

2. System Overview

For the IWSLT 2014 evaluation campaign, we build a phrase-based statistical machine translation system that is based on a log-linear discriminative model.

2.1. SMT System

Our phrase-based statistical machine translation system is mainly based on the work of an open-source toolkit Moses [5]. A number of widely used features are adopted in our SMT system, including bidirectional phrase translation probabilities and lexical translation probabilities, language model, word penalty, phrase penalty, distance-based distortion model, and hierarchical reordering model [6].

We use a modified GIZA++ toolkit for word alignment, which extend the IBM models and HMM model by the addition of an l_0 prior to the word-to-word translation model. It can reduce overfitting, and generate less useless phrase pairs. We test different heuristics (*grow*, *grow-diag-final*, *grow-diag-final-and*) for symmetrizing bidirectional word alignment results. For different tasks, there are some notable differences in performance among heuristics. When calculating the phrase translation probabilities, we use Good-Turing smoothing techniques, rather than using relative frequency. It turned out to be useful to improve translation performance.

Since the SMT system is based on a log-linear model, feature weights have a big impact on translation quality.

While tuning feature weights, we tried different development sets. In addition, tuning algorithm also makes some difference. We tested MERT and kbMIRA, and found that kbMIRA is better than MERT in our experiments.

N-gram language models are created with the SRILM toolkit [7]. We evaluate the tokenized translation results in case-sensitive fashion, using the BLEU metric [8].

For date, time and other number related expressions (DTN), we have some special treatments. We firstly write some rules to identify DTN expressions in source language, and then edit corresponding translations in target language for each identification rule. Regular expressions are used for the task. Finally, these rules with translations are added into the translation model with high translation probabilities.

Some source words, which cannot be translated by the translation model, are called out-of-vocabulary (OOV) words. We make additional process for two kinds of OOV words. The first case is those do occur in the TED training corpus, but no corresponding translations in the phrase table due to the restriction of phrase extraction. In this case, we make use of lexical translation table to translate these OOV words. The second case is those do not occur in TED corpus but appear in MultiUN corpus. For these words, we extract their translation from the MultiUN phrase table. In the other cases, we simply drop OOV words.

To exploit some features that are not suitable to be added into the decoder, we use them in the reranking step. The n-best translation results are generated by the decoder, and then additional feature scores are calculated for each hypothesis. Finally, the n-best list is reranked according to the new feature set.

Along with the techniques mentioned above, we also implement some novel models to further improve translation performance, which are described as follows.

2.2. Language Model

We put an emphasis on language modeling. Besides the 5-gram model trained from TED corpus, we also train several n-gram language models from the English Gigaword corpus and News Crawl corpora. Each of them is taken as a separate feature in the log-linear model. In addition, we build several other types of language models described below.

2.2.1. Backward Language Model

We build a backward n-gram language model [9], where the probability of each word is estimated depending on words following it:

$$P(W) = \sum_{i=T}^{1} P(w_i \mid w_{i+1}, w_{i+2}, ..., w_{i+n-1}) \quad (1)$$

We use the model in reranking stage. In our experiments, the backward language model can sometimes be helpful, but not always.

2.2.2. Class-based Language Model

Data sparseness is a common problem in natural language processing. Automatically clustering words from monolingual or bilingual training corpora into word classes is a widely used method to improving statistical models [10]. Here we build a class-based language model, and find it helpful in improving translation quality.

Firstly, we made use of `mkcls` in Moses toolkit to train a mapping from each word to a fixed class. Then we project words in training corpus to classes and train a class-based language model. In our system, a 7-gram class-based model is trained using SRILM toolkit. Class-based language model probability is used as a separate feature in decoder.

2.2.3. Recurrent Neural Network Language Model

Recent work has shown that recurrent neural network language models outperform significantly the n-gram models, even in case when n-gram models are trained on much more data. Moreover, when compared to feed-forward neural network language model, the RNNLM allows effective processing of sequences and patterns with arbitrary length, and it enables to learn long-distance dependence in the hidden layer.

In our system, we use the open-source RNNLM toolkit [11] to train a recurrent neural network language model. The model is used at the reranking stage to generate an additional feature for each hypothesis.

2.3. Neural Network Joint Model

Neural network based technologies are playing a more and more important role in recent natural language processing research. Recent studies on machine translation, which introduce neural network language model (NNLM) as features, turns out to be a breakthrough progress [12]. Moreover, some researchers present a novel formulation of a neural network joint model (NNJM) [13] as an extension of NNLM, which introduces dependence on source words. Though NNJM is just based on a lexicalized probabilistic model and a simple feed forward neural network, the experimental results show that it has significant improvements over the baseline systems.

The basic NNJM (s2t.l2r) formula can be written as:

$$P(T \mid S) \approx \prod_{i=1}^{|T|} P(t_i \mid t_{i-1}, ..., t_{i-n+1}, \xi_i) \quad (2)$$

where T is the target sentence, S is the source sentence, ξ_i is the source word window. In this circumstance, each target word t_i is affiliated with exactly one source word at index a_i. Then ξ_i is a m-word source window centered at a_i.

$$\xi_i = s_{a_i-(m-1)/2}, ..., s_{a_i}, ..., s_{a_i+(m-1)/2} \quad (3)$$

By changing the dependence order among target words, or swapping source and target languages, we can implement several variants of NNJM (s2t.r2l, t2s.l2r, t2s.r2l) as shown in Equation 4 to 6, where ζ_i is similar with ξ_i, which is just a replacement of source word s into target word t.

$$P(T \mid S) \approx \prod_{i=1}^{|T|} P(t_i \mid t_{i+1}, ..., t_{i+n-1}, \xi_i) \quad (4)$$

$$P(S \mid T) \approx \prod_{i=1}^{|S|} P(s_i \mid s_{i-1}, ..., s_{i-n+1}, \zeta_i) \quad (5)$$

$$P(S \mid T) \approx \prod_{i=1}^{|S|} P(s_i \mid s_{i+1}, ..., s_{i+n-1}, \zeta_i) \quad (6)$$

As the computational cost of NNLMs is a significant issue in decoding phase, we adopt two techniques for speeding up NNJM computation: self-normalization and pre-computation.

The self-normalization technique aims to avoid computing output softmax over the entire target vocabulary. Mainly, it replaces the training objective function with

$$L = \sum_i [\log(P(x_i)) - \alpha \log^2(Z(x_i))] \quad (7)$$

where $Z(x)$ is the summing part of softmax normalizer, and α is the parameter that controls trade-off between neural network accuracy and mean self-normalization error. At decoding phase, we simply use the input value of output layer as feature score, rather than $\log(P(x))$.

Another technique is called pre-computation, which computes dot product between the projection layer (word embedding) and the first hidden layer in advance. Furthermore, the computation of hyperbolic tangent (*tanh*) can also be accelerated using a lookup table.

In our experiments, we integrate the NNJM s2t.l2r model into our decoder, and the other variant models are used in the reranking step.

2.4. Domain Adaptation

Besides the TED portion data, the MultiUN [14] bilingual data can also be used for building translation models. However, the MultiUN Chinese-English parallel corpus provided by the IWSLT2014 Evaluation Campaign is aligned in chapter level. It cannot be used directly. To solve the problem, we firstly employ a tool hunalign [15] to automatically align the corpus at sentence-level.

In addition, the MultiUN data is almost 50 times larger than the in-domain parallel data, so it is unwise to treat them equally. We adopt a cross entropy based text selection method to choose partial volume from the MultiUN data [16]. In this method, an in-domain language model is applied to calculating cross entropy for each sentence pair, and then those with relatively low cross entropy are selected.

We select about 20% portion of the MultiUN data, and divide these data into several groups. For each group, we can train a translation model. There are two ways to incorporate these translation models into the system: linear interpolation and log-linear interpolation. We use the simple yet effective linear interpolation method. Each component probability in the translation model is linearly interpolated together. For example, let us consider the "backward" probability $p(s|t)$ of source language phrase *s* being generated by target language phrase *t*. For a set of $p_i(s|t)$, each trained on a sub-corpus, the mixture model is computed as

$$p(s|t) = \sum_{i=1}^{N} \alpha_i p_i(s|t) \quad (8)$$

To set the weights α_i, we firstly extract a set of phrase pairs from an in-domain development set using the training procedure. This yields a joint distribution $\tilde{p}$, which is used to define a maximum likelihood objective function as in Equation 9. The weights can then be learned efficiently using EM algorithm, which was first proposed in [17].

$$\tilde{\alpha} = \arg\max_{\alpha} \sum_{s,t} \tilde{p}_i(s,t) \log \sum_{i}^{N} \alpha_i p_i(s|t) \quad (9)$$

3. Experiments

In this section, we describe the experimental setup and results for both Chinese-English and English-Chinese translation tasks. We use the IWSLT 2013 test set for evaluating the techniques described above.

3.1. Chinese-English

As preprocessing, all the English texts in the corpora were tokenized by the tokenization tool in Moses toolkit. All Capital letters were converted to lower case. For Chinese, sentences need to be split into words. We compared several Chinese word segmentation tools and finally chose the in-house implementation. As post-processing, we use an SMT-based recaser to restore the true case for the output of the decoder. The experimental results are given in Table 1. All scores are case-sensitive BLEU.

3.1.1. Baseline Systems

Firstly, we built a weak baseline system ("*weak-baseline*" in Table 1) with the similar setup to that of the official baseline system in IWSLT 2013 [18]. All models are trained using the in-domain TED data provided by the campaign [19]. Bidirectional word alignments were trained by GIZA++ and symmetrized using *grow-diag-final-and* heuristic. An MSD-based lexical reordering model was applied. A 5-gram language model with modified Kneser-Ney smoothing was trained from the English part of the parallel corpus using SRILM toolkit. The weights of all features are optimized on dev2010 using MERT. Translation quality was evaluated on the tst2013 set in IWSLT 2013.

We obtained the strong baseline system by improving the following components: development set, word alignment, translation model, reordering model and weight tuning algorithm.

The official website released four sets for tuning, which are dev2010, tst2010, tst2011, and tst2012. Since bigger development set showed better performance in our pilot experiments, we combined them together and formed a big development set. Using the big development set for weight tuning gave rise to an improvement of +0.4% BLEU ("bigdev" in Table 1).

For word alignment, we improved GIZA++ with the l_0-norm. Although it has almost no effect on tst2013, it improved the development set by +0.16% BLEU. So we still keep it in our system. By simply replacing *grow-diag-final-and* by *grow*, our system gained further +0.14% BLEU.

There are only 180k sentence pairs in the TED training corpus, which is quite small. Over 90% phrase pairs in the phrase table occurred only once in the training corpus. This indicates data sparseness. Similarly to language model smoothing, we applied Good-Turing [20] to smoothing occurrence counts of phrase pairs, instead of using the counts directly. We obtained an improvement of +0.33% BLEU with Good-Turing smoothing ("GT smoothing" in Table 1).

As for the MSD based lexical reordering model, it is known that there are inconsistence about reordering orientation detection between training and decoding time [21]. A simple yet effective improvement is the hierarchical reordering model (HRM). Replacing MSD by HRM gave us another gain of +0.29% BLEU.

Finally, we adopted kbMIRA instead of MERT to tune feature weights. kbMIRA optimize BLEU less aggressively, improving model score and BLEU correlation across range of hypothesis. It produced an additional gain of +0.3% BLEU. Now we denote the system as "*strong-baseline*" in Table 1.

From "*weak-baseline*" to "*strong-baseline*", there are totally improvements of +1.45% BLEU on tst2013. Base on the "*strong-baseline*", we further improve our system by

adding more language models, neural network joint model, domain adapted translation models, etc.

3.1.2. Additional Features

Besides the parallel corpora, the official website also provides a number of monolingual English data. We used them to train n-gram language models. To be specific, each corpus was used to train a 5-gram language model with modified Kneser-Ney smoothing. Then we selected top ten language models according to the perplexity of LM on development set. Table 2 shows all of the selected corpora and the corresponding perplexities. The TED in-domain language model was the primary LM used in baseline systems and naturally has the lowest perplexity. We added these ten out-of-domain LMs to the decoder as separate features, and tuned their weights together with other features. We were surprising to see that these ten LMs gave us a great improvement up to +1.88% BLEU, which is the biggest improvement among all the techniques.

For NNJMs, we set up a projection layer of 192 dimensions and single hidden layer of 512 dimensions. Sizes of both input and output vocabularies are 10K. During training we set an initial learning rate of 10^{-3} and a mini-batch size of 128. Training was performed on GPU processor, and the decoding was carried out on CPU. By incorporating the s2t.l2r model into decoder, we achieved further gain of +0.5% BLEU.

MultiUN is the only out-of-domain parallel data that can be used in the campaign. It contains 9.5 million sentences, which is 52 times larger than the in-domain data. Instead of using all the MultiUN data, we selected about 1.9M parallel sentences from it using a cross-entropy based method [16], and divided them into four groups (125K, 250K, 500K, 1000K sentence pairs for each group). From each group, we trained one translation model. Then we linearly interpolated these models together with the in-domain model. Interpolation weights were trained by EM algorithm. This domain adaptation method improves performance by +0.18% BLEU (denoted by "*+UN_DA*").

In the last step, we tried to use more features to rerank k-best translations. We firstly generate 1000 best hypotheses from the "*+UN_DA*" system. Then five additional features were added for each hypothesis: three NNJM model (s2t.r2l, t2s.l2r, t2s.r2l) scores, a RNNLM score and a backward language model score. kbMIRA was used to tune weights for all features including those used in decoding. Reranking brought a further improve of +0.22% BLEU. The "reranking" result was our primary submission.

Table 1: Results for Chinese-English MT task

system	dev	tst2013
weak-baseline	10.61	14.19
+bigdev	13.20	14.59
+l_0-norm	13.36	14.58
+grow	13.42	14.72
+GT smoothing	13.65	15.05
+HRM	13.87	15.34
+ kbMIRA (strong-baseline)	13.91	15.64
+10 LMs	15.44	17.52
+NNJM	16.01	18.02
+UN_DA	16.20	18.20
+reranking	16.42	18.42

Table 2: Selected corpora for LMs and corresponding perplexities

data	bigdev
WIT[3] mono English (in-domain)	95.0
CzEng 1.0 from WMT14	103.7
News Crawl: 2013 from WMT14	104.8
News Crawl: 2012 from WMT14	107.4
News Crawl: 2011 from WMT14	108.9
nyt_eng from gigaword fifth edition	109.0
News Crawl: 2009 from WMT14	113.1
News Crawl: 2008 from WMT14	114.2
ltw_eng from gigaword fifth edition	116.8
News Crawl: 2010 from WMT14	117.4
News Crawl: 2007 from WMT14	128.6

Table 3: Results for English-Chinese MT task

System	bigdev	tst2013
	BLEU (char-based)	**BLEU** (char-based)
weak-baseline	14.92	18.87
strong-baseline	20.03	21.46
+wcLM	20.36	21.70
+OSM	20.47	22.05
+NNJM	20.83	22.35
+UN_DA	20.91	22.44
+reranking	21.01	22.55

3.2. English-Chinese

For the English-Chinese MT task, all the parallel and monolingual data are preprocessed exactly the same way as the Chinese-English task. All the scores showed in Table 3 are char-based BLEU. We also trained a weak baseline and a strong baseline using the same techniques as those in the Chinese-English task. The development set is also the same one, except that the source and target language are reversed. The "*strong-baseline*" achieves an improvement of +2.59% BLEU on tst2013 over the "*weak-baseline*".

Then, we improved the "*strong-baseline*" system by adding a 7-gram word class language model into the decoder (wcLM, +0.24% BLEU). All words were classified into 400 classes. After that, an Operation Sequence Model (OSM) was added. It gains +0.35% BLEU (Theses two techniques were also tried on the Chinese-English task, but no improvements were achieved. So we neglect them in the above sub-section). We also adopted NNJM (s2t.l2r, +0.31% BLEU) and domain adaptation for translation models (UN_DA, +0.09% BLEU). Finally, we reranked 1000-best hypotheses generated by "*+UN_DA*" system (reranking, +0.11% BLEU). The "reranking" result was our primary submission.

4. Conclusions

In this paper, we presented our submission runs and technical details of the IWSLT 2014 Evaluation Campaign in the optional MT track on Chinese-English and English-Chinese translations. The baseline system utilizes a state-of-the-art phrase-based translation decoder. After applying a lot of novel models and techniques, the translation results were significantly improved.

To summarize, main improvements result from the following techniques:

- Rich language model features. We build several large language models and integrate them into the log-linear model as separate features. We build different types of language models such as RNNLM, class-based LM and reverted-directional LM.
- Successfully implemented neural network models. We build NNJM, RNNLM for decoding or reranking, and achieve significant improvements.
- Effectively used data. We make a big development set by combining several previous test sets. Bigger development set produces better results. We extract some useful texts from MultiUN, which helps improve the translation model.

In the future, we are planning to integrate more features into our log-linear models.

5. References

[1] F. J. Och, "Minimum Error Rate Training in Statistical Machine Translation," in Proc. of the 41th Annual Meeting of the Association for Computational Linguistics (ACL), Sapporo, Japan, July 2003, pp. 160–167.

[2] A. Vaswani, L. Huang, and D. Chiang, Smaller alignment models for better translations: unsupervised word alignment with the l_0-norm. In Proc. ACL, 311–319, 2012.

[3] C. Cherry and G. Foster. 2012. Batch tuning strategies for statistical machine translation. In HLT-NAACL, pages 427–436, Montr´ eal, Canada, June.

[4] Nadir Durrani, Helmut Schmid, and Alexander Fraser. 2011. A Joint Sequence Translation Model with In tegrated Reordering. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, pages 1045–1054, Portland, Oregon, USA, June.

[5] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: open source toolkit for statistical machine translation. In Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL) on Interactive Poster and Demonstration Sessions, pages 177-180.

[6] M. Galley and C. D. Manning, "A simple and effective hierarchical phrase reordering model," in Proceedings of the Conference on Empirical Methods in Natural Language Processing, ser. EMNLP '08. Stroudsburg, PA, USA:Association for Computational Linguistics, 2008, pp. 848–856.

[7] A. Stolcke, "SRILM – An Extensible Language Modeling Toolkit," in Proc. of the Int. Conf. on Speech and Language Processing (ICSLP), vol. 2, Denver, CO, Sept. 2002, pp. 901–904.

[8] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a Method for Automatic Evaluation of Machine Translation," in Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics, Philadelphia, Pennsylvania, USA,July 2002, pp. 311–318.

[9] Deyi Xiong, Min Zhang, Haizhou Li: Enhancing language models in statistical machine translation with backward n-grams and mutual information triggers. ACL-HLT 2011: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics, Portland, Oregon, June 19-24, 2011; pp.1288-1297.

[10] J. Wuebker, S. Peitz, F. Rietig, and H. Ney, "Improving statistical machine translation with word class models," in Conference on Empirical Methods in Natural Language Processing, Seattle, USA, Oct. 2013, pp. 1377–1381.

[11] Tomas Mikolov, Stefan Kombrink, Lukas Burget, JH Cernocky, and Sanjeev Khudanpur. 2011. Extensions of recurrent neural network language model. In 2011 IEEE International Conference on Acoustics, Speech and Signal Processing, pages 5528-5531.

[12] Holger Schwenk. 2010. Continuous-space Language Models for Statistical Machine Translation. Prague Bull. Math. Linguistics, 93:137-146.

[13] Jacob Devlin, Rabih Zbib, Zhongqiang Huang, Thomas Lamar, Richard Schwartz, and John Makhoul. 2014. Fast and Robust Neural Network Joint Models for Statistical Machine Translation. In 52nd Annual Meeting of the Association for Computational Linguistics, pages 1370-1380.

[14] A. Eisele and Y. Chen, "MultiUN: A Multilingual Corpus from United Nation Documents," in Proceedings of the Seventh conference on International Language Resources and Evaluation, May 2010, pp. 2868–2872.

[15] D. Varga, L. Nemeth, P. Halacsy, A. Kornai, Viktor Tron, and V. Nagy. 2005. Parallel corpora for medium density languages. In RANLP, pages 560–596, Borovets, Bulgaria.

[16] A. Axelrod, X. He, and J. Gao, "Domain adaptation via pseudo in-domain data selection," in Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, Edinburgh, Scotland, UK., July 2011, pp. 355–362.

[17] George Foster, Cyril Goutte and Roland Kuhn. 2010. Discriminative Instance Weighting for Domain Adaptation in Statistical Machine Translation. EMNLP. Cambridge, MA.

[18] M. Cettolo, J. Niehues, S. Stker, L. Bentivogli, and M. Federico, "Report on the 10th iwslt evaluation campaign," in Proceedings of the 10th International Workshop on Speech Language Translation, 2013.

[19] M. Cettolo, C. Girardi, and M. Federico. 2012. WIT3: Web Inventory of Transcribed and Translated Talks. In Proc. of EAMT, pp. 261-268, Trento, Italy.

[20] G. Foster, R. Kuhn, and H. Johnson. "Phrasetable smoothing for statistical machine translation". Proc. EMNLP, pp. 53-61, Sydney, Australia, July 2006

[21] C. Tillmann. 2004. "A Unigram Orientation Model for Statistical Machine Translation". NAACL.