The MIT-LL/AFRL MT System

Wade Shen, Brian Delaney, and Tim Anderson

MIT Lincoln Laboratory 244 Wood St. Lexington, MA 02420 {swade,bdelaney}@ll.mit.edu	Air Force Research Laboratory 2255 H St. Wright-Patterson AFB, OH 45433 Timothy.Anderson@wpafb.af.mil

Abstract

The MIT-LL/AFRL MT system is a statistical phrase-based translation system that implements many modern SMT training and decoding techniques. Our system was designed with the long term goal of dealing with corrupted ASR input for Speech-to-Speech MT applications. This paper will discuss the architecture of the MIT-LL/AFRL MT system, and experiments with manual and ASR transcription data that were run as part of the IWSLT-2005 Chinese-to-English evaluation campaign.[1]

1. Introduction

In recent years, the development of statistical methods for machine translation has resulted in high quality translations that can be used in real applications with increasing confidence. Specific advancements include:

- Extracting word alignments from parallel corpora [1] [2]
- Learning and modeling the translation of phrases [4] [5]
- Combining and optimizing model parameters [6] [7] [8]
- Decoding and rescoring techniques [9] [10]

Our system draws from these advances and implements a number of these techniques including log-linear model combination and minimum error rate training to translate foreign language sentences. We developed our system during preparation for IWSLT-2005 to serve as a platform for future research. Most of the components of our system have been developed in-house in order to facilitate future experimentation.

In subsequent sections, we will discuss the details translation system including our alignment and language models and methods we've implemented for optimization and decoding. The basic translation training and decoding processes are shown in Figure 1. We start with a word alignment extracted from a training set using GIZA++. These alignments are expanded and phrases are counted to form the phrase translation model. Language models are then trained from the English side of training set (and possibly with other English texts, if available).

Using development bitexts separated from the training set, we then employ a minimum error rate training process to optimize model parameters utilizing a held out development set. These trained parameters and models can then be applied to test data during decoding and rescoring phases of the translation process.

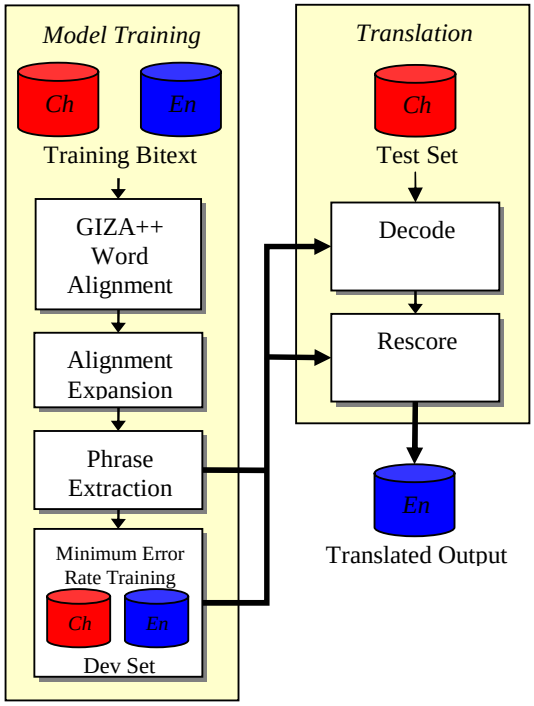

Figure 1: Basic Statistical Translation Architecture

2. Translation Models

The basic phrase-translation model we employ is described in detail [4] and [5]. We use GIZA++ [1], [3] and [4] to generate alignments between foreign language and English language words/tokens in both directions (f-to-e and e-to-f). An expanded alignment is then computed using heuristics that interpolate between the intersection and union of these bidirectional word alignments as detailed in [4] and [5]. Then phrases are extracted from the expanded alignments to build the phrase model. We introduced two minor modifications to this basic process in our system:

1. Prior to iterative expansion of intersection alignment, we first add points that are unaligned in both the source and target language sentences that otherwise fit the standard inclusion criteria.

[1] *This work is sponsored by the United States Air Force Research Laboratory under Air Force Contract FA8721-05-C-0002. Opinions, interpretations, conclusions and recommendations are those of the author and are not necessarily endorsed by the United States Government.*

2. We introduced a target phrase length factor during phrase extraction that limits target phrases to *Length(source) * TargetLengthFactor*.

The combination of these two enhancements provided a two point improvement in BLEU score on the IWSLT-2004 development set in the supplied data condition.

We extract translation models for both translation directions (i.e. *P(f|e)* and *P(e|f)*). In addition to these models, we add lexical weights extracted from the expanded alignment process in both translation directions [5] and a fixed phrase-penalty [11].

3. Distortion Model

For this evaluation we used a simple distortion model that was described by Koehn et al in [5] and shown here:

$$P_D(e \mid f) = \exp(-\sum_i \left| (FinalW_{i-1} + 1) - FirstW_i \right|) \quad (1)$$

where $FinalW_{i-1}$ is the position of the final word of the previous phrase and $FirstW_i$ is position of the first word of the current phrase. Although simple, we found this model to be very effective for the Chinese-to-English task. Additional constraints on distortion can be chosen to limit the phrase reordering during search, however we chose not to limit distortion when decoding on the IWSLT05 test data.

4. Language Models

We used the SRI language modeling toolkit to build language models [12]. A trigram language model is used during initial decoding. Two additional language models are introduced during rescoring and minimum error rate training: a four-gram language model, and a 5-gram class-based language model. All of these models were trained with modified Knesser-Ney interpolation as suggested by Goodman and Chen in [13] and [14].

5. Minimum Error Rate Training

Our system employs minimum error rate training to optimize parameter weights for each of the individual model components that are summarized in table 1. Many different optimization strategies have been proposed for combination of log-linear models (see [6], [7], [8] and [15] for further details). The algorithm we used is a variation of the method described by Och in [7]. In particular, we optimize each of these parameters assuming log-linear combination as given by

$$\hat{P}(e \mid f) = \frac{\exp(\sum_{\forall M} \lambda_M h_M(f, e))}{\sum_{\forall e'} \exp(\sum_{\forall M} \lambda_M h_M(f, e'))} \quad (2)$$

where λ_M represents individual model weights and $h_M(f, e)$ represents the feature function associated with estimates from each individual translation/language model.

The optimization process attempts to sets parameter weights that minimize the overall error rate. To do this, sentences from a development set are used to sample the error surface defined by the model parameters listed above and the loss function to be minimized. Each sentence is decoded to produce n-best translation output samples and a line search is then performed per sentence/model parameter to minimize the overall error rate.

This process yields a set of parameter weights for each model that can be then used for decoding and rescoring. The training is iterative in that n-best lists are created using the new parameters and merged with the existing n-best lists to increase the resolution of our sampling for each iteration. As reported by Och [7], this iterative procedure converges after 6-8 iterations.

	Model Weight Parameters
1	*P(f|e)* – Forward Translation Model
2	*P(e|f)* – Backward Translation Model
3	*LexW(f|e)* – Forward Lexical Weight
4	*LexW(e|f)* – Backward Lexical Weight
5	*PPen* – Constant, per-phrase Penalty
6	*WPen* – Constant, per-word Penalty
7	*Dist* – Distortion Model
8	*Tri-LM* – Trigram Language Model
9	*4-LM* – Four-gram Language Model
10	*ClassLM* – Five-gram class-based LM

Table 1: Model Parameters used for minimum error rate training

6. Decoding

For the IWSLT 2005 results that were submitted, we used Philip Koehn's Pharaoh Decoder [9]. Pharaoh is stack decoder with an A-star search heuristic. We have recreated similar results with our own in-house decoder, which was not ready before the submission deadline.

Our in-house decoder utilizes a Viterbi graph search algorithm, where the graph is built from left to right a word at a time. Each node in the search graph contains a list of back-pointers to previous nodes, the probabilities for each translation option, the trigram/bigram context needed for node expansion, and the best path so far. When creating new nodes, the number of nodes can be greatly reduced by using bigrams whenever a particular trigram does not exist in the language model. After all possible phrases are added for each word, both beam pruning and histrogram pruning are used to rid the search graph of unlikely candidate nodes based on the best path.

This search algorithm offers fast decoding and easy generation of output word lattices by simply traversing the final data structure. In the case of monotone decoding, this search algorithm is capable of real-time decoding one word at a time (as from a speech recognizer).

In order to include distortion in the search, additional information needs to be kept for each node including a word coverage vector, the distortion probability, and an estimate of the future cost. Nodes are connected to previous nodes only if the intersection of the word coverage vectors is empty. The

future cost estimate is used so that all nodes can be pruned together (A-star search). This limits the search space enough so that unconstrained reordering can be used without running out of memory, but there is a possibility that the search will not be able to select a final path that covers all input words. In the case of a search failure, the search is restarted using improved future cost heuristics from the previous pass.

7. Results

In this section, we present results from applying the system described above to the IWSLT 2005 evaluation. We present results from both manual and ASR transcription conditions in the supplied data track for Chinese-to-English translation.

Because the training data in this track is limited to 20,000 sentence pairs, we employed a number of small modifications to the basic phrase-model extraction procedure to minimize the number of out-of-vocabulary tokens during decoding. These techniques are described in the section below.

For both ASR and manual transcription conditions we did not limit distortion and the four-gram and class-based language models were applied during rescoring. For efficiency reasons we limited the decoder stack to 350 hypotheses.

7.1. Dev Set Experiments

Two development sets were provided as part of the IWSLT 2005 evaluation campaign:

- **Devset 1:** Evaluation data from CSTAR 2003
- **Devset 2:** Evaluation data from IWSLT 2004

We used these tests to construct 3 other dev sets for more robust parameter optimization:

- **Devset 3:** first ½ of Devsets 1 and 2
- **Devset 4:** second ½ of Devsets 1 and 2
- **Devset 5:** sum of Devset 1 and 2

For minimum error rate training we held out one of devsets 1-4 and tested against another devset (without overlap). Interestingly, devset sentences were generally longer in set 4 and, not surprisingly, scores were generally worse. Devset 5 was used to optimize final parameters for test set decoding. Tables 2 and 3 shows scores from different development run configurations.

Test \ *Dev*	*1*	*2*
1		36.64
2	42.00	

Table 2: Devset 1 and 2 Results

Test \ *Dev*	*3*	*4*
3		42.44
4	33.84	

Table 3: Devset 3 and 4 Results

Using devset 2 for development testing, we experimented with a number of different configurations of our basic SMT system. Table 4 shows the effect of different preprocessing options and training parameters on BLEU scores for devset 2.

Configurations	BLEU
Baseline: char seg., UTF-8, 4x TPF, no opt (hand-tuned weights)	39.12
+ lexbackoff	40.32
+ lexbackoff + 2x TPF	40.76
+ lexbackoff + 2x TPF + Word Seg.	34.12
+ lexbackoff + 2x TPF + Optimization	40.99
+ lexbackoff + 2x TPF + extra LMs	41.45
+ lexbackoff + 2x TPF + extra LMs + Optimization	**42.00**

Table 4: Model Parameters used for minimum error rate training (TPF = Target Phrase Factor devset 1 was used for minimum error rate training).

Applying these settings and the optimization weights used for devset 2, we tested against each of the ASR outputs for devset 2. Table 5 shows results from each of ASR engine and their corresponding ASR for both 1-best and N-best ASR hypotheses:

ASR	N-best Length	N-best Correct %	BLEU
1	1	68.7	26.15
2	1	80.9	32.30
3	1	87.3	35.08
1	20	80.1	28.37
2	20	91.8	36.90
3	20	94.5	37.68

Table 5: Devset 2 results with various ASR transcripts.

For each of the n-best > 1 conditions we simply decode the entire ASR n-best list and merge results before rescoring. We did not weight ASR acoustic or language models for this experiment, but this remains to be explored in future experiments.

7.2. IWSLT 2005 Manual Transcription

We submitted results using multiple optimization parameters from different devsets (devset 5 parameters being primary). These results are shown in Table 6. As expected, optimizing with devset 5 yielded the best performance. Although our optimization loss function was based on BLEU-4, our system performed reasonably well with other metrics as well (METEOR, PER, NIST). It is also interesting to note that the additional 506 sentences of the CSTAR-03 development set did not provide us much gain over the IWSLT-04 set alone.

Better language model rescoring seems to have made the biggest difference. As we did not fully examine language model training possibilities, it is possible that further improvements in this area may yield even better performance.

Configurations	NIST	BLEU
Devset 1 optimization	9.296	44.58
Devset 2 optimization	9.307	44.83
Devset 5 optimization (w/o additional language models)	9.151	43.44
Devset 5 optimization	**9.311**	**44.96**

Table 6: Test set scores with various optimization settings

7.3. IWSLT 2005 ASR Transcription

Our primary submission for this track used n-best output from the ASR system. As we saw with our devset 2 experiments, n-best ASR output gives a gain of 2-3 BLEU points over the 1-best hypothesis.

Configurations	N-best Length	NIST	BLEU
Devset 1 optimization	1	7.211	32.28
Devset 2 optimization	1	7.208	32.27
Devset 5 optimization	1	7.222	32.40
Devset 1 optimization	20	7.633	35.62
Devset 2 optimization	20	7.557	35.67
Devset 5 optimization	**20**	**7.555**	**35.96**

Table 7: Test set scores with various optimization settings

Again in this condition, our system's performance was quite reasonable, especially with respect to non-BLEU metrics. We expect that more efficient use of ASR model parameters (acoustic and language model scores, and perhaps lattice posteriors) we could yield even better performance.

8. Discussion

From experiments with both the development and test data, it is clear that better language modeling can increase the performance of phrase-based statistical machine translation systems as evidence from [16] and [17] also suggests. It is interesting to note that in the case of Chinese translation, distortion models seems to play a major role and, as such, further work is needed to design models that account for the reordering effects we see in Chinese. In past experiments, this has not necessarily been true, for instance with French to English translation monotone decoding often yields better performance.

We are just beginning to explore methods for fusing ASR and MT systems. From an anecdotal evaluation of our ASR MT results by a native speaker, we found that only 2 of 30 sampled sentences contained additional errors (relative to MT output from manual transcription). Both this and objective evaluation results from this evaluation are encouraging, but it is clear that much more work is needed to make the output of Speech MT usable. We expect that joint optimization of ASR model parameters and MT model parameters could yield better results but careful research is needed to integrate ASR and MT parameter appropriately.

9. Acknowledgements

We would like to thank John Luu at AFRL for his error analysis efforts after the evaluation. We would also like to thank Doug Jones for setting up an in-house MT evaluation test suite for benchmarking our development progress. Finally, thanks to the members of Lincoln Lab IST group for leaving us plenty of machines to run our test data.

10. References

[1] Brown, P., Della Pietra, V., Della Pietra, S. and Mercer, R. "The mathematics of statistical machine translation: Parameter estimation", *Computational Linguistics* 19(2):263--311. 1993.

[2] Vogel, S., Ney, H., and Tillmann, C. "HMM-based word alignment in statistical translation", In *Proceedings of the 16th Conference on Computational Linguistics - Volume 2*, pp. 836-841, Copenhagen, Denmark, August, 1996.

[3] Al-Onaizan, Y., Curin, J., Jahr, M., Knight, K., Lafferty, J., Melamed, I.D., Och, F.J., Purdy, D., Smith, N.A., Yarowsky, D., "Statistical machine translation: Final report", In *Proceedings of the Summer Workshop on Language Engineering*. John Hopkins University Center for Language and Speech Processing, Baltimore, MD 1999.

[4] Och, F. J. and Hermann, N. "Improved Statistical Alignment Models", In *Proc. of the 38th Annual Meeting of the Association for Computational Linguistics*, pp. 440-447, Hong Kong, October, 2000.

[5] Koehn, P., Och, F. J. and Marcu, D., "Statistical Phrase-Based Translation", In *Proceedings of the Human Language Technology Conference 2003 (HLT-NAACL 2003)*, Edmonton, Canada, May 2003.

[6] Och, F. J. and Ney, H., "Discriminative Training and Maximum Entropy Models for Statistical Machine Translation". In *ACL 2002: Proc. of the 40th Annual Meeting of the Association for Computational Linguistics*, pp. 295-302, Philadelphia, PA, July 2002.

[7] Och, F. J., "Minimum Error Rate Training for Statistical Machine Translation", In *ACL 2003: Proc. of the 41st Annual Meeting of the Association for Computational Linguistics*, Japan, Sapporo, July 2003.

[8] Venugopal, A. and Vogel S., "Considerations in Minimum Classification Error and Maximum Mutual Information Training for Statistical Machine Translation", In *Proceedings of the Tenth Conference of the European Association for Machine Translation (EAMT-05)*, Budapest, Hungary, May 2005.

[9] Koehn, P., "Pharaoh: A Beam Search Decoder for Phrase-Based Statistical Machine Translation Models", In *Proceedings of the Association of Machine Translation in the Americas (AMTA-2004)*, Washington, DC, October, 2004.

[10] Ueffing, N., Och, F. J., Ney, H., "Generation of Word Graphs in Statistical Machine Translation", In *Proc. Conference on Empirical Methods for Natural Language Processing*, pp. 156-163, Philadelphia, PA, July 2002.

[11] Thayer, I., Ettelaie, E., Knight, K., Marcu, D., Munteanu D. S., Och F. J., and Tipu, Q., "The ISI/USC MT

System", In *Proc. of the International Workshop on Spoken Language Translation*, Kyoto, Japan, 2004.

[12] Stolcke, A., "SRILM – An Extensible Language Modeling Toolkit", In *Proceedings of the International Conference on Spoken Language Processing*, Denver, CO, 2002.

[13] Chen, S. F. and Goodman, J., "An Empirical Study of Smoothing Techniques for Language Modeling", *Computer Speech and Language*, 13:359–394, October, 1999.

[14] Goodman, J., "A Bit of Progress in Language Modeling", *Computer Speech and Language*, pp. 403—434, 2001.

[15] Cettolo, M. and Federico, M., "Minimum Error Rate Training of Log-Linear Translation Models", In *Proc. of the International Workshop on Spoken Language Translation*, Kyoto, Japan, 2004.

[16] Bender, O., Zens, R., Matusov, E. and Ney, H., "Alignment Templates: the RWTH SMT System", In *Proc. of the International Workshop on Spoken Language Translation*, Kyoto, Japan, 2004.

[17] Och, F., "The Google MT System", Presentation at *NIST MT Workshop*, Bethesda, MD, 2005.