ISCA Archive
http://www.isca-speech.org/archive

International Workshop on Spoken Language Translation (IWSLT) 2006
Keihanna Science City, Kyoto, Japan
November 27-28, 2006

The MIT-LL/AFRL IWSLT-2006 MT System

Wade Shen, Brian Delaney†

MIT Lincoln Laboratory
244 Wood St.
Lexington, MA 02420, USA
swade@ll.mit.edu

Tim Anderson

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 and limited amounts of training data for speech-to-speech MT applications. This paper will discuss the architecture of the MIT-LL/AFRL MT system, improvements over our 2005 system, and experiments with manual and ASR transcription data that were run as part of the IWSLT-2006 evaluation campaign.

1. Introduction

In recent years, the development of statistical methods for machine translation has made usable MT a real possibility. Specifically, advances in methods to:

- Extract word alignments from parallel corpora [1][2]
- Learn and model the translation of phrases [3] [4]
- Combine and optimize model parameters [5] [6] [7]
- Decode and Rescore Test data [8] [9]

These advances have helped to dramatically increase the quality of MT output. Our 2006 IWSLT system extends these methods and work we did in 2005 [10].

In subsequent sections, we will discuss the details of the translation system including our alignment and language models and methods we've implemented for optimization and decoding. Specifically, we will highlight improvements and changes made to:

1. Better utilize the larger 2006 training set
2. Coverage of Italian and Japanese
3. Enhance the coverage of extracted phrases

†This work is sponsored by the Air Force Research Laboratory under Air Force contract FA8721-05-C-0002. Opinions, interpretations, conclusions and recommendations are those of the authors and are not necessarily endorsed by the United States Government.

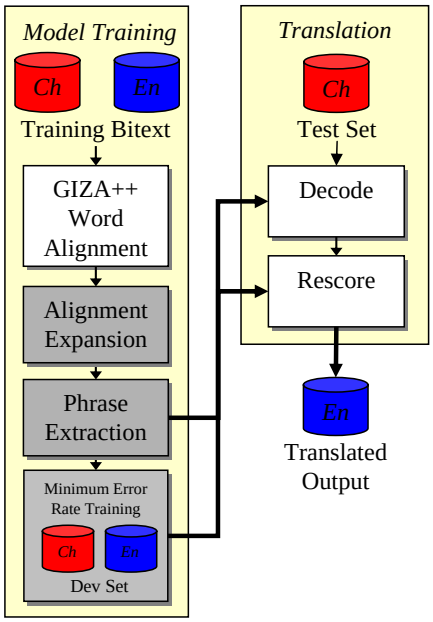

Figure 1: *Basic Statistical Translation Architecture*

4. Better models and better decoding
5. Increase gains from rescoring n-best lists

As this year's evaluation conditions have changed, our basic translation training and decoding processes have been adapted accordingly, as shown in Figure 1. Boxes in grey have not changed substantially since 2005. Refer to [10] for more detail regarding the implementation of these modules.

We submitted systems for Chinese, Japanese and Italian-to-English language pairs. In each case, we used only the supplied data for each language pair for training and optimization. From these data, we extract word/character alignments. These alignments are then expanded using slightly modified versions of standard heuristics. This process is described in detail in Section 3. Phrases are then extracted and counted, and the resulting phrase table is then used for de-

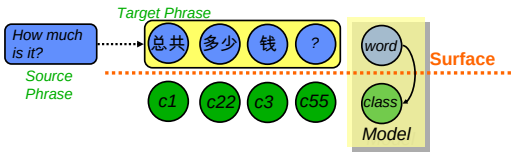

Figure 2: *A Factor-based Consistency-Checking Model*

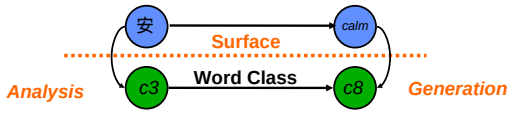

Figure 3: *A Parallel Word Class/Surface Translation Model*

coding and rescoring. Language models are trained using the English side of each language pair.

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.

2. Data Preprocessing

For Chinese and Japanese texts, we used the supplied UTF-8 encodings and converted all roman characters into ASCII. We used Latin-1 encoding for all Italian texts. Source and target side training texts are lower-cased before training.

Because this year's evaluation data (and devset 4) included no source punctuation, we implemented a source-language repunctuator to better match the training data.

3. Improved Word/Character Alignments

In this year's system we employed multiple word and character alignment strategies, extending the method described in [11]. For all language pairs, we combine alignments from IBM model 5 see [1] and [12] and alignments extracted using the competitive linking algorithm (CLA) described in [13]. We apply a simple χ^2 likelihood function, though we found only minor differences between this function and others that have been proposed in the literature [14]. Phrases were extracted from both types of alignments and combined in one phrase table. This was done by summing counts of phrases extracted from alignment types before computing the relative frequency used in the our phrase tables.

Additionally, for Chinese-to-English translation, both word and character segmentation were for training CLA and GIZA alignment models. Phrases were then extracted from all four alignments and combined. Word segmented phrases were resegmented into characters before counting.

4. Improved Translation Models

Following the 2006 JHU summer workshop we conducted a number of experiments with factored translation models using our training/decoding paradigm. To this end we integrated the `moses` decoder into minimum error rate training decoding processes. This allowed us to try two different factor-based approaches to the IWSLT Chinese-English translation task.

Factored translation models extend standard phrase-based statistical models by representing words as vectors of *factors*. This representation can be used to decompose words into constituent parts (e.g. lemma + affix) for the purpose of modeling them separately, or as generalizing words into larger linguistic "classes" (e.g. part-of-speech). From a factored representation, it is possible to train standard statistical models that are then combined using standard log-linear assumptions in which feature functions of the form $h_{FACTOR_k}(e_{1...i}, f_{1...j})$ represent translation likelihoods that are specific to factor k and special generation features $h_{gen}(FACTOR_k(e_i), FACTOR_l(e_i))$ that represent the likelihood of generating $FACTOR_k$ from $FACTOR_l$.

Because we did not have access to analysis tools in Chinese during the IWSLT evaluation, we chose to create models using automatically derived word classes (as generated by `mkcls`). In our experiments words are represented both by their surface form and by their associated word classes.

Using this representation we trained two different models:

- *Consistency-Checking Model* – Translate source surface forms to target, generate word classes for each target, then apply a class-based LM.
- *A Parallel Translation Model* – Translate both source surface forms and word-classes to target word/class pairs, then apply a class-based LM.

These models are shown schematically in Figure 2 and Figure 3, respectively. We note that the parallel approach is quite similar to the alignment template model proposed in [15] with an additional surface-to-surface form translation model. These models were not applied in time for official submission to the 2006 evaluation, but in post-evaluation experiments we found these models to be quite helpful.

5. Improved Decoding

For the 2006 evaluation we used a combination of two decoders: our in-house decoder `mtdecoder` and the `moses` decoder developed as part of the 2006 JHU summer workshop. For most experiments, both decoders performed on par with each other (though we generally used our own decoder for minimum error rate training, because of it's speed). For factored experiments, we used `moses`. With both decoders we found it advantageous to use 4-gram and 5-gram language models in decoding. Our official submissions for Chinese, Japanese and Italian use 4-gram Interpo-

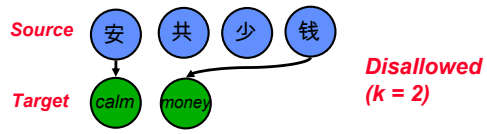

Figure 4: *An example of a disallowed reordering using IBM constraints*

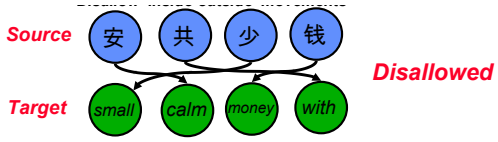

Figure 5: *An example of a disallowed reordering using ITG constraints*

lated Knesser-Ney models trained using the SRI Language Modeling Toolkit [16] [17] [18].

Using our decoder we implemented three types of reordering constraints, revisiting work done in [19] with the IWSLT-2006 data. We explored both ITG [20] and IBM constraints, and the results shown in Section 7 indicate that different reordering constraints don't decrease the BLEU score significantly in most language pairs while reducing decoding time by 20-50%. Both constraints disallow certain reording configurations. Figures 5 and 4 offer examples of these configurations. Details of these experiments are described in [21].

6. Rescoring N-best Lists

As in 2005, we employ minimum error rate training to optimize model scaling factors for both decoding and rescoring features. In this year's evaluation, we added 5-gram rescoring language models and 6-gram class-based rescoring language models after decoding. After the evaluation we added sentence length posterior features for rescoring. A full list of the feature functions used in our system is shown in Table 1.

We approximate sentence length posteriors from the n-best list as:

$$P(L|f_{1\ldots J}) \approx \sum_{\{e \mid |e|=L\}} P(e_{1\ldots L}|f_{1\ldots J}) \qquad (1)$$

Similarly IBM model 1 scores can be computed for each n-best list entry:

$$P_{ibm1} \approx \frac{1}{(I+1)^J} \prod_{j=1}^{J} \sum_{i=1}^{I} p(f_j|e_i) \qquad (2)$$

7. Development Experiments

In preparation for the arrival of the official evaluation data, we conducted experiments with our system using `dev4` in each of the language pairs. For these experiments we set aside `dev1` for minimum error rate training.

[1] features added after the official submission

Decoding Features
$P(f\|e)$
$P(e\|f)$
$LexW(f\|e)$
$LexW(e\|f)$
Phrase Penalty
Lexical Backoff
Word Penalty
Distortion
$\hat{P}(e)$ – 4-gram language model
Rescoring Features
$\hat{P}(e)$ – 5-gram LM
$\hat{P}(e)$ – 6-gram class-based LM
$P_{Model1}(f\|e)$ – IBM model 1 translation probabilities
Sentence-length posterior[1]

Table 1: *Feature functions used in the translation model*

7.1. Segmentation and Alignment

For different language pairs we employ different segmentation techniques. We use basic word segmentation for Italian, combining phrases extracted from IBM model 5 alignments with CLA alignments. For Japanese, we found it optimal to use word segmentation with character segmentation backoff with CLA alignments. In this configuration, words that were unseen in training (OOV) are broken into constituent characters then translated using character phrases. In the Chinese case, we use both word and character segmentation. From both, we compute both CLA and IBM model 5 alignments and extract phrases that are then normalized to character segmentation when aggregating counts.

Tables 2, 3 and 4 show a summary of results for various configurations of segmentation and alignment.

Configuration	*BLEU*
Character Segmented	21.24
Word Segmented	21.01
Char+Word Segmented	21.21
Char+Word Segmented + CLA	**22.18**

Table 2: *Segmentation/alignment results for Chinese (*`dev4`*)*

7.2. Rescoring

In addition to standard features that we use during decoding, we introduce a number of additional features for rescoring n-best lists generated by our decoder (or `moses`). For the 2006 evaluation we tried a number of new features, including longer context LMs (text and class-based), IBM model 1, unigram posteriors and sentence length posteriors. Empir-

Configuration	*BLEU*
Word Segmented	23.63
Word Segmented + Character Backoff	23.82
Word Segmented + CLA	23.34
Word Segmented + Character Backoff + CLA	**24.28**

Table 3: *Segmentation/alignment results for Japanese (*`dev4`*)*

Configuration	*BLEU*
Word Segmented	35.13
Word Segmented + CLA	**37.40**

Table 4: *Segmentation/alignment results for Italian (*`dev4`*)*

ically, we found that all features with the exception of unigram posteriors were beneficial. As shown in Table 5 rescoring is helpful when testing on `dev4` for all language pairs, though it varies widely (from 3.32% to 10.76% relative improvement).

	BLEU		
Configuration	*Chinese*	*Japanese*	*Italian*
Baseline 4-gram Decode	21.39	21.92	36.92
w/5-gram rescore LM	21.55	–	–
w/6-gram class-based LM	21.52	–	–
w/Model 1	21.86	–	–
w/Sent. Length Posterior	22.10	–	–
ALL Features	**22.10**	**24.28**	**37.40**

Table 5: *Rescoring results for all languages (*`dev4`*)*

7.3. Pre/Post-Processing

During the evaluation, we explored different pre and post-processing options to optimize this year's official evaluation criterion (mixed-case, with punctuation, no source punctuation provided). We tried two different methods of producing target punctuation: 1) training asymmetric models by removing source punctuation from train and development corpora, and 2) repunctuating source sentence in the supplied development and test corpora.

To produce mixed-case output, we applied implemented an HMM-based truecasing model as proposed in [22]:

$$w*_{1\ldots j} = \arg\max_{w_{1\ldots j}} P(w_{1\ldots j}|s_{1\ldots j}) \quad (3)$$

$$= \arg\max_{w_{1\ldots j}} P(s_{1\ldots j}|w_{1\ldots j}) \quad (4)$$

$$*P(w_{1\ldots j}) \quad (5)$$

where a standard, interpolated language model approximation is used as in:

$$\hat{P}(w_{1\ldots j}) \approx \prod_{k=1}^{j} P(w_k|w_{k-1}\ldots w_{k-n+1}) \quad (6)$$

and an approximate table of conditonal emission probabilities is represented by:

$$\hat{P}(s_{1\ldots j}|w_{1\ldots j}) \approx \prod_{k=1}^{j} P(s_k|w_k) \quad (7)$$

Where $w*_{1\ldots k}$ is the maximum likelihood TrueCased output sequence and $s_{1\ldots j}$ is the corresponding lower-case input. As shown in Table 6, automatic repunctuation of the input source is beneficial in performance terms. Similarly, small gains can be had by choosing the appropriate language model order for TrueCasing.

	BLEU		
Configuration	*Chinese*	*Japanese*	*Italian*
Remove Source Punctuation			
w/4-gram TrueCase LM	21.86	23.14	36.64
Repunctuate Source			
w/3-gram TrueCase LM	21.93	–	–
w/4-gram TrueCase LM	**22.10**	**24.28**	**37.40**
w/5-gram TrueCase LM	22.10	–	–

Table 6: *Effects of different pre/post-processing methods (*`dev4`*)*

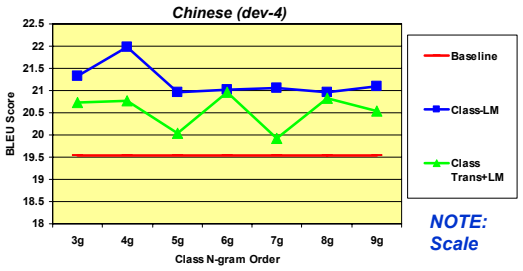

Figure 6: *Performance of two factored models with different class-LM contexts*

7.4. Factored Models

After the official evaluation deadline, we ran a number of experiments to explore the performance of the factored models described in Section 4. Our experiments focus on a baseline Chinese-to-English system trained using only word segmentation and optimized as described above. Due to time constraints, we did not perform the rescoring described in Section 7.2. With this configuration, our baseline system achieve a BLEU score of 19.60 on `dev4` with the official evaluation criteria.

We ran experiments with both *Consistency Checking* models using a class-based language model, and *Parallel*

Translation models using both class-based translation and language models. As shown in Figure 6, both factored approaches achieve substantial gains, though the *Consistency Checking* model (shown as Class-LM) is consistently better than both the baseline and the *Parallel Translation* model (shown as Class Trans+LM). This approach equals the performance of our best rescoring model on `dev4` despite starting from a worse baseline.

We have seen that limitations in the current implementation of `moses` may cause search errors in our parallel translation models. Despite current limitations, our parallel models offer some advantage.

7.5. Decoder Reordering Constraints

	BLEU/Time (secs)		
Configuration	*Chinese*	*Japanese*	*Italian*
free	20.32/3509.5	22.35/3309.7	35.85/90.6
IBM	19.85/2961.0	21.46/2969.3	35.52/36.2
ITG	19.85/2961.0	21.37/1868.7	35.52/36.2

Table 7: *Performance of different reordering constraints (*`dev4`*)*

Although we did not use ITG or IBM reordering constraints in our official submissions, some development experiments with these constraints did yield gains. Unfortunately, these gains were not consistent across dev sets. Table 7 shows the performance of different reordering constraints in constrast to our baseline configuration, free reordering, in which all possible reorderings are allowed within a fixed window (in our default configuration this is set to 10).

Gains in processing time are quite apparent. 20-60% improvement in speed can be had with minimal BLEU score impact using these reordering constraints. More detailed experiments with these constraint can be found in [21].

8. Evaluation Results and Analysis

Text Input	*BLEU*		
Configuration	*Chinese*	*Japanese*	*Italian*
Opt. (`dev4`)	**21.57**	**20.99**	**35.74**
Opt. (`dev1`)	20.66	20.24	34.40
Opt. (`dev4`) – No Rescoring	21.27	–	–

Table 8: *Overall performance of submitted systems with text input (*`test-2006`*)*

Tables 8 and 9 show our official submissions to the 2006 IWSLT evaluation. Official primary submissions are shown in bold. Each primary system performed well, ranking 3rd/4th in ASR BLEU scores and 2nd/4th in text BLEU scores among submitted systems. Note that our primary system was not always best (e.g. Italian ASR condition). Our primary submissions were optimized using `dev4`. These submissions processed 1-best ASR input and reference transcription. Our secondary submissions decoded 10-best from the ASR lattice, merging MT n-best lists and rescoring with ASR features as described in [10].

ASR Input	*BLEU*			
Configuration	*Ch (Read)*	*Ch (Spon.)*	*Jp*	*It*
1-best, Opt (`dev4`)	**18.61**	**16.57**	**18.91**	**27.98**
10-best, Opt (`dev4`)	17.42	16.57	–	28.81
1-best, Opt (`dev1`)	18.46	–	18.43	27.64

Table 9: *Overall performance of submitted systems with ASR input (*`test-2006`*)*

Reruning our system using the 2005 train/dev/test paradigm, we found that our system gained over 4 BLEU points (8.7% relative improvement) with respect to our previous best.

Our next steps include further development of our in-house decoder and experiments with factored models using better baselines and better search methods.

9. Acknowledgements

We'd like to thank Eric Chapla, Johnn Luu, Tyler Pierce for providing language support in Japanese and Chinese respectively. We would also like to thank the staff of the Information Systems and Technology group at MIT Lincoln Lab for making machines available for this evaluation effort.

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, 1996.

[3] 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, 2000.

[4] 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, 2003.

[5] 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.

[6] 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, 2003.

[7] 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, 2005.

[8] 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, 2004.

[9] 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, 2002.

[10] Shen, W., Delaney, B. and Anderson, T., "The MIT-LL/AFRL MT System," In Proc. Of the International Workshop on Spoken Language Translation, Pittsburgh, PA, 2005.

[11] Chen, B. et al, "The ITC-irst SMT System for IWSLT-2005," In Proc. Of the International Workshop on Spoken Language Translation, Pittsburgh, PA, 2005.

[12] 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.

[13] Melamed, D., "Models of Translational Equivalence among Words," Computational Linguistics, vol. 26, no. 2, pp. 221-249, 2000.

[14] B. Chen and M. Federico, "Improving Phrase-based Statistical Translation Through Combination of Word Alignments," FinTAL 2006, LNAI 4139, pp. 356-367, 2006.

[15] F. J. Och, C. Tillmann, H. Ney, "Improved Alignment Models for Statistical Machine Translation," Proc. of the Joint Conf. of Empirical Methods in Natural Language Processing and Very Large Corpora; University of Maryland, College Park, MD, 1999.

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

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

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

[19] Zens, R., and Ney, H., "A Comparative Study on Reordering Constraints in Statistical Machine Translation," In Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL), Sapporo, Japan, 2003.

[20] Wu, D., "Stochastic inversion transduction grammars and bilingual parsing of parallel corpora," Computational Linguistics, 1997.

[21] Delaney, B., Shen, W., and Anderson, T., "An Efficient Graph Search Decoder for Phrase-Based Statistical Machine Translation," Submitted to the International Workshop on Spoken Language Translation, Kyoto, Japan, 2006.

[22] Lita, V., et al, "tRuEcasIng," In Proceedings of the 41st Annual Meeting of the Association of Computational Linguistics, Sapporo, Japan, 2003.