The JHU Workshop 2006 IWSLT System

Wade Shen†

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

Richard Zens

Lehrstuhl für Informatik 6
Computer Science Department
RWTH Aachen University
D-52056 Aachen, Germany
zens@cs.rwth-aachen.de

Nicola Bertoldi, Marcello Federico

ITC-irst
Centro per la Ricerca
Scientifica e Tecnologica
I-38050 Povo (Trento), Italy
{federico|bertoldi}@itc.it

Abstract

This paper describes the SMT we built during the 2006 JHU Summer Workshop for the IWSLT 2006 evaluation. Our effort focuses on two parts of the speech translation problem: 1) efficient decoding of word lattices and 2) novel applications of factored translation models to IWSLT-specific problems. In this paper, we present results from the open-track Chinese-to-English condition. Improvements of 5-10% relative BLEU are obtained over a high performing baseline. We introduce a new open-source decoder that implements the state-of-the-art in statistical machine translation.

1. Introduction

With advances in both speech recognition and language translation technologies, speech translation has become more viable for real applications. It is still, however, the case that both ASR and MT technologies are prone to high levels of error. Thus, robust techniques of combining these technologies are needed to make speech translation viable for real applications. Empirically, we know that reduced error rates from upstream ASR systems result in higher translation quality. Our goal is to exploit multiple hypothesis to effictively *lower* the error rate relative to the 1-best transcription.

In this paper, we introduce the notion of confusion network MT decoding for ASR input and discuss novel applications of this technique. We also present a new application of factored translation models for postprocessing MT output.

Both confusion network decoding and factored models are implemented in a new, open-source MT decoder called `moses` that was built as part of the 2006 JHU summer workshop. We introduce `moses` as a platform for MT research.

1.1. Confusion Networks

During decoding, ASR systems can typically generate multiple alternatives in addition to a single-best transcription. It is customary to represent these alternatives using a word graph or word lattice that contains information about the word sequences as well as start and end times. Figure 1 shows an example of a typical word lattice for a sentence of 15 words.

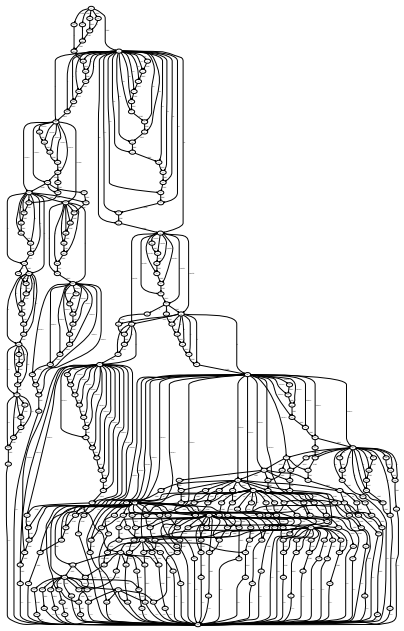

Figure 1: *An Example ASR Word Lattice*

Ideally, it would be possible to process such a word lattice as input into a machine translation system so as to consider all possible translation options, not just the single best hypothesis. In practice, direct MT decoding without heavy pruning of ASR word lattices can increase the computational complexity of the decoding problem beyond current computing capabilities. Some efforts toward direct lattice decoding have shown success [1] but require careful preprocessing of input lattices or severe pruning during search.

To simplify the problem, it is useful to note that for machine translation tasks, time information (i.e. word start and end times) is not needed and, in fact, this information greatly increases the size of the search problem during decoding, es-

†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.

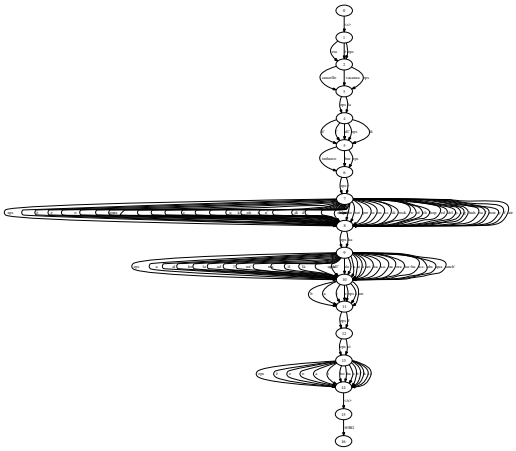

Figure 2: *An example ASR confusion network*

era 0.997	cancello 0.995	ϵ 0.999	di 0.615	imbarco 0.999	...
è 0.002	vacanza 0.004	la 0.001	d' 0.376	bar 0.001	
ϵ 0.001	ϵ 0.002		l' 0.002		
			...		
			ϵ 0.001		

Figure 3: *ASR confusion network in tabular form*

pecially when arbitrary reordering is allowed. Removing this information, it is possible to then approximate a word lattice using a confusion network [2].

The confusion network is a linearization of the word lattice that combines (through a set of heuristics) paths with different word start and end times while preserving all paths through the word lattice. The confusion network forces word alternatives into strict slots but allows `NULL` transitions (represented by ϵ). In each slot, word alternatives are labeled with posterior probabilities computed from ASR acoustic and language model features. Figures 2 and 3 shows a graphical depiction of a typical confusion network.

In prior work [3], it was shown that the algorithm for decoding of confusion networks is a direct generalization of the standard phrase-based MT decoding strategy [4]. In the `moses` decoder, we extend this algorithm using an efficient prefix-tree representation of the phrase table. Details of this are discussed in Section 3.

1.2. Factored Translation Models

Factored translation models extend standard phrase-based translation models by incorporating multiple levels of linguistic representation, or factors. Factored models decompose the translation process into multiple translation and generation subprocesses at different linguistic levels. Factored models allow, for instance, the translation of part-of-speech tags in addition to the surface form as shown graphically in Figure 4. In this case, both source input and target translations are represented using two factors: surface form and POS. In this model, the translation process can be expressed as:

Figure 4: *An example factored translation model*

$$\begin{aligned}\log P(\vec{e}|\vec{f}) &\propto \sum_{\forall r} \lambda_r h_r(\vec{f},\vec{e}) \\ &\propto \sum_{\forall r \in surface} \lambda_r h_r(surface(\vec{f}), surface(\vec{e})) \\ &\quad + \sum_{\forall s \in POS} \lambda_s h_s(POS(\vec{f}), POS(\vec{e})) \\ &\quad + \sum_{\forall g \in Gen} \lambda_g h_g(surface(\vec{e}), POS(\vec{e}))\end{aligned}$$

Using this decomposition we define *translation* processes (e.g. $\sum_{\forall s \in POS} \lambda_s h_s(\vec{f},\vec{e})$) and *generation* processes (e.g. $\sum_{\forall g \in Gen} h_g(surface(\vec{e}), POS(\vec{e}))$) that mutually constrain each other. In our current implementation of `moses`, generation processes are limited to a single target word context.

In this model the standard phrase-base approach can be seen as a special case of more general factored models.

For the IWSLT 2006 evaluation we did not use deep linguistic analysis for our submitted systems, but we did make use of the factored translation facility within the `moses` decoder. In Section 4 we discuss novel applications of factored translation models for the IWSLT 2006 translation task.

2. Model Training and Optimization

We used Chinese-to-English models trained by MIT Lincoln Laboratory (both phrase tables and language models) and applied MIT-LL's preprocessing to all development and test data. Phrase tables were trained using only the provided training sets. The training procedure is outlined below:

- Generate word and character segmented forms of the training corpus
- Compute GIZA++ and Competitive Linking (CLA) alignments for both word and character segmented data [6] [7]
- Extract phrases for all variants of the training corpus
- Split word-segmented phrases into characters
- Combine phrase counts from all variants and normalize

- Train language models using target language side of the training corpus
- Train TrueCase models
- Train source language repunctuation models

Features
$P(f\|e)$
$P(e\|f)$
$LexW(f\|e)$
$LexW(e\|f)$
Phrase Penalty
Word Penalty
Distortion
$P(e)$ - 4-gram language model

Table 1: *Features from training*

Using the extracted phrase table we apply a minimum error rate procedure based on [8]. We used either development set 1 (the evaluation set from CSTAR-2003) or development set 4 to optimize model scaling factors (λs) for each of the features shown in Table 1. For our primary systems, no rescoring features were added, though significant improvements might be had through n-best list reranking features [5].

More details of the training procedures can be found in [5].

2.1. ASR Lattice Preprocessing

For data in both read and spontaneous speech conditions, we converted the provided word lattices into confusion networks using the SRI `lattice-tool` [9] without pruning. Columns for source punctuation were inserted by aligning the 1-best transcription with columns of the confusion network and then introducing posterior probabilities for all possible punctuation marks.

For text condition data, we used two different forms of preprocessing:

- Repunctuate the source string using the 1-best punctuation hypothesis
- Create a confusion network of punctuation by inserting *all* possible punctuation marks (and ϵ) between each word.

Section 5 describes experiments we ran with these configurations in detail.

3. Decoder Implementation

As part of the 2006 JHU Summer Workshop we extended the `moses` decoder to process input confusion networks.

A key insight is that, due to their linear structure, confusion network decoding is very similar to text decoding. During the decoding, we have to look up the translation options of spans, with a span being a contiguous sequence of source positions. The main difference between confusion network and text decoding is that in text decoding there is exactly one source phrase per span, whereas in confusion network decoding there can be multiple source phrases per span. In fact, in a confusion network the number of source phrases per span is exponential in the span length. The exact number is the product of the column depths over the positions in the span. A naive approach to generate the translation options per span is to enumerate the source phrases of the span and to look up the translation options of each source phrase in the phrase table. For obvious reasons this is inefficient.

A more efficient implementation can be achieved if we use a prefix tree representation for the source phrases in the phrase table and generate the translation options incrementally over the span length. Thus, when looking up a span (j_1, j_2), we can exploit our knowledge about the span $(j_1, j_2 - 1)$. Thus, we have to check only for the known prefixes of $(j_1, j_2 - 1)$ if there exists a successor prefix with a word in column j_2 of the confusion network. If all the word sequences in the confusion network also occur in the phrase table, this approach still enumerates an exponential number of phrases. Though worst case complexity is still exponential in the span length, this is unlikely to happen in practice. In our experiments, we do not observe the exponential behavior. What we observe is a constant overhead compared to text input.

4. Novel Applications of Factored Models

As previously described, the `moses` decoder supports decoding of factored translation models. For this year's IWSLT evaluation we applied factored translation models to the problem of TrueCasing MT output.

We apply a simple HMM model for truecasing [11] implemented using the `disambig` tool developed by SRI [9]. Our model can be defined as follows:

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

where $s_{1\ldots j}$ is the sequence of uncased input words and $w^*_{1\ldots j}$ is the maximum-likelihood output truecased sentence. The posterior distribution $P(w_{1\ldots j}|s_{1\ldots j}, \lambda)$ can be decomposed as:

$$\begin{aligned} P(w_{1\ldots j}|s_{1\ldots j}) &= \frac{P(s_{1\ldots j}|w_{1\ldots j}) * P(w_{1\ldots j})}{P(s_{1\ldots j})} \\ \arg\max_{w_{1\ldots j}} P(w_{1\ldots j}|s_{1\ldots j}) &= \arg\max_{w_{1\ldots j}} P(s_{1\ldots j}|w_{1\ldots j}) \\ &\quad * P(w_{1\ldots j}) \end{aligned}$$

where we approximate each distribution as:

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

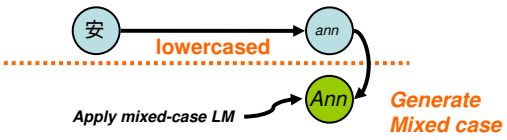

Figure 5: *Integrated TrueCasing Model*

	Chinese	English
sentences	40 K	
running words	351 K	365 K
avg. sent. length	8.8	9.1
vocabulary entries	11 K	10 K

Table 2: Corpus statistics for the Chinese-English task.

and

$$\hat{P}(s_{1\ldots j}|w_{1\ldots j}) \approx \prod_{k=1}^{j} P(s_k|w_k) \qquad (2)$$

We noticed that this model could be represented using a factored translation model in which mixed-case output is generated from a latent lower-case factor. In this way, it is possible to incorporate the TrueCasing process into translation decoding directly. The TrueCasing process can also be represented as components of the log-linear translation model. As such it is possible to to jointly optimize the truecase and translation models. Figure 5 shows a schematic of the integrated truecasing model we used for IWSLT-2006. In this model lower-cased source text is translated into a lowercased target form. Then a TrueCased surface form is generated and constrained by a mixed-case language model. In this way, our model implements equations 1 and 2 directly into the decoding process. Specifically, equation 1 is implemented as a surface language model, and the HMM transition distribution described in equation 2 is implemented as a generation step.

5. Results

The experiments were carried out on the *Basic Travel Expression Corpus* (BTEC) task [10]. This is a multilingual speech corpus which contains tourism-related sentences similar to those that are found in phrase books. The corpus statistics are shown in Table 2. For the supplied data track, 40 000 sentences of training corpus and three test sets were made available for each language pair.

5.1. Chinese-to-English

In this section, we will present the experimental results for the Chinese–English task. The statistics of the confusion networks are summarized in Table 3. Note that the average length of the sentences in the `dev4` test set is about twice as large as in the training data. We also present the depths of the confusion networks. On average we have between two and three alternatives per position. At some positions, however, there are more than 90 alternatives.

	speech type	
	read	spontaneous
avg. length	17.2	17.4
avg. / max. depth	2.2 / 92	2.9 / 82
avg. number of paths	10^{21}	10^{32}

Table 3: Confusion network statistics for the `dev4` set (489 sentences).

	speech type	
	read	spontaneous
`dev4`	12.8%	21.9%
`test`	15.2%	20.6%

Table 4: 1-best character error rates (CER) for `dev4` and `test` sets (489 and 500 sentences respectively).

In Table 5, we present the translation results for the Chinese–English task for different input conditions on the `dev4` and the `eval` test sets. All scores reported here use two-pass TrueCasing and source repunctutaion. Comparing the translation results of 1-best and confusion network, we observe a small but consistent improvement for read speech. For spontaneous speech, the improvements are larger, e.g. 1.1% BLEU for the `eval` test set.

As described in Section 2.1, we did some experiments on devset 4 using the confusion network decoder for repunctuation. The results are presented in Table 6. We observe a small improvement when using a confusion network for the punctuations. Note that this system was not optimized, so we might expect larger improvement when optimizing for this type of input.

Similar experiments were carried out comparing integrated TrueCasing to a standard 2-pass procedure. After optimization, the integrated model performs slightly better than the baseline. These results are shown in Table 7.

		speech type	
test set	input	read BLEU [%]	spontaneous BLEU [%]
`dev4`	verbatim	21.4	
	1-best	19.0	17.2
	full CN	19.3	17.8
`eval`	verbatim	21.4	
	1-best	18.5	17.0
	full CN	18.6	18.1

Table 5: Chinese–English: translation results for different input types.

punctuation input type	BLEU [%]
1-best	20.8
confusion network	21.0

Table 6: Chinese–English: translation results for punctuation insertion (devset 4).

TrueCase Method	BLEU [%]
Standard Two-Pass: SMT + TrueCase	20.65
Integrated Factored Model (optimized)	21.08

Table 7: Chinese–English: translation results for different TrueCasing configurations (devset 4).

6. Conclusions

For the official evaluation, our ASR and text input systems achieved state-of-the-art performance. In this work we described two novel, statistical approaches to the speech translation problem: 1) confusion network decoding and 2) a factored decomposition of standard phrase-based models. Confusion network decoding of the ASR input provided the largest gain (6.5% relative) on this task and we expect that these gains will carry over to other languages and other speech translation tasks.

Although our factored model did not show gain in preliminary experiments, following results in [5], we suspect gains are possible.

7. Acknowledgements

We would like to thank our JHU summer workshop team members (Philipp Koehn, Hieu Hoang, Chris Dyer, Ondrej Bojar, Chris Callison-Burch, Brooke Cowan, Christine Moran, Alexandra Constantin and Evan Herbst) who made this construction of this system possible. We wish to acknowledge their diligent efforts to make the `moses` decoder stable in a six-week period.

We would also like to thank the staff and faculty of CLSP at John's Hopkins University for graciously hosting us during the summer workshop.

8. References

[1] L. Mathias and W. Byrne "Statistical Phrase-based Speech Translation", Submitted to IEEE Trans. Speech and Audio Proc., 2006.

[2] L. Mangu, E. Brill, and A. Stolcke, "Finding consensus in speech recognition: word error minimization and other applications of confusion networks," Computer Speech and Language, pages 373-400, 2000.

[3] N. Bertoldi and M. Federico, "A new decoder for spoken language translation based on confusion networks," in IEEE ASRU Workshop, 2005.

[4] P. Koehn, "Pharaoh: a beam search decoder for phrase-based statistical machine translation models," In Proceedings of AMTA, 2004.

[5] A. Mauser, R. Zens, E. Matusov, S. Hasan, and H. Ney, "The RWTH Statistical Machine Translation System for the IWSLT 2006 Evaluation," Submitted to the International Workshop on Spoken Language Translation, 2006, Kyoto, Japan.

[5] W. Shen, B. Delaney and T. Anderson, "The MIT-LL/AFRL IWSLT 2006 Translation System," Submitted to the International Workshop on Spoken Language Translation, 2006, Kyoto, Japan.

[6] B. Chen, R. Cattoni, N. Bertoldi, M. Cettolo, M. Federico, "The ITC-irst SMT System for IWSLT-2005", In Proc. Of the International Workshop on Spoken Language Translation, Pittsburgh, PA, 2005.

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

[8] F. J. Och, "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.

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

[10] T. Takezawa, E. Sumita, F. Sugaya, H. Yamamoto, and S. Yamamoto, "Toward a broad-coverage bilingual corpus for speech translation of travel conversations in the real world," in Proc. of the Third Int. Conf. on Language Resources and Evaluation, Las Palmas, Spain, May 2002, pp. 147–152.

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