The University of Washington Machine Translation System for IWSLT 2006

Katrin Kirchhoff[*], *Kevin Duh*[*], *Chris Lim*[†]

Department of Electrical Engineering[*]
Department of Computer Science and Engineering[†]
University of Washington, Seattle, USA
{katrin,duh}@ee.washington.edu, chrislim@cs.washington.edu

Abstract

This paper describes the University of Washington's submission to the IWSLT 2006 evaluation campaign. We present a multi-pass statistical phrase-based machine translation system for the Italian-English open-data track. The focus of our work was on the use of heterogeneous data sources for training translation and language models, the use of several novel rescoring features in the second pass, and exploiting N-best information for translation in the ASR-output condition. Results show mixed benefits of adding out-of-domain data and using N-best information and demonstrate improvements for some of the novel rescoring features.

1. Introduction

We present a two-pass statistical phrase-based machine translation system developed for the IWSLT 2006 evaluation. For this task we concentrated on a single language pair, Italian-English, and on the correct transcription and ASR-output conditions. We used the ASR output provided and did not produce our own ASR hypotheses from the raw speech data. Since the BTEC task is a sparse-data task, our focus for this evaluation was on exploring the use of heterogeneous data sources for training. In addition, we investigated several novel features for rescoring and the use of N-best information for the ASR-output condition. This paper is structured as follows: we first describe the data sources and preprocessing used. Sections 4 and 5 describe first-pass hypothesis generation and second-pass rescoring. Postprocessing and spoken-language specific processing are presented in Sections 6 and 7. We then present experiments and the official evaluation results. Section 10 describes additional analyses performed after the official evaluation and Section 11 concludes.

2. Data

The UW system participated in the open data track. For training we used the BTEC Italian-English training data provided for this evaluation campaign, along with the devset1, devset2, and devset3, resulting in approximately 190K words of in-domain training data (including punctuation). In addition, we used the publicly available Europarl corpus of Italian/English [1] for training the translation model. This corpus is very different from BTEC in that it contains edited transcriptions of parliamentary proceedings; thus, the domain differs from that of a travel task, and the style is that of written text. The size of the Europarl corpus is approximately 17M words. We also used the Fisher corpus for training certain second-pass language models. The Fisher corpus is a collection of English conversational telephone speech covering a variety of speakers and topics. It consists of approximately 2.3M word tokens.

All development/evaluation was done on devset4, since it was expected to be most similar to the test data. This set was randomly split into a development set of 350 sentences and a held-out set of 139 sentences.

3. Preprocessing

The BTEC data was preprocessed by first segmenting lines with multiple sentences into single sentences according to the punctuation. Punctuation was then removed and all words were lowercased. The Europarl data was also lowercased and sentence pairs with a length ratio greater than 9 were removed. For the evaluation system, no additional preprocessing was performed. After the official evaluation we attempted to improve the use of the Europarl data by modifying the English side to render it more similar to spoken language style and modifying the Italian side to more closely match the transcription conventions used in the BTEC corpus. All English sentence punctuation was removed, common English contractions were added to 90% of the corpus, and abbreviations or titles were punctuated. For example, the sentence *thank you , mr segni , i shall do so gladly .* became *thank you mr. segni i'll do so gladly*. In the Italian text, sentence punctuation was also removed, apostrophes denoting contractions were joined to the preceding part of the word, and common abbreviations were expanded. For example, the sentence ... *condannare l' arresto della sig.ra gladys marin ed esigerne l' immediata scarcerazione ?* became ... *condannare l' arresto della signora gladys marin ed esigerne l' immediata scarcerazione*. However, these modification did not affect translation performance significantly.

4. First-Pass Translation System

We use a multi-pass statistical phrase-based translation system based on a log-linear probability model:

$$e^* = argmax_e p(e|f) = argmax_e \{\sum_{k=1}^{K} \lambda_k \phi_k(e,f)\} \quad (1)$$

where e is and English and f a foreign sentence, $\phi(e,f)$ is a feature function defined on both sentences, and λ is a feature weight. The first pass generates up to 2000 translation hypotheses per sentence using the public-domain Pharaoh decoder [2] and a combination of the following nine model scores:

- two phrase-based translation scores
- two lexical translation scores
- data source indicator feature
- word transition penalty
- phrase penalty
- distortion penalty
- language model score

The first two of these are explained below (Section 4.1). The word transition and phrase penalty are constant weights added for each word/phrase used in the translation, thus controlling the length of the translation. The distortion penalty assigns a weight proportional to the distance by which phrases are reordered during decoding; here, the distortion penalty is constant since monotone decoding is used and no reordering is allowed. (Initial experiments show that monotone decoding outperforms non-monotone decoding.) Weights for these scores are optimized using the minimum-error rate training procedure in [3]. The optimization criterion is the BLEU score on the development set as defined above (Section 2). The second pass rescores the first-pass output with additional, more advanced model scores. A post-processing step is then performed to restore true case and punctuation.

4.1. Translation Model

The translation model is defined over a segmentation of source and target sentence into phrases: $f = \bar{f}_1, \bar{f}_2, ..., \bar{f}_M$ and $e = \bar{e}_1, \bar{e}_2, ..., \bar{e}_M$. Phrase pairs of up to length 7 are extracted from the training corpus which was previously word-aligned using GIZA++. The extraction method is the technique described in [4] and implemented in [2]: the corpus is first aligned in both translation directions, the intersection of the alignment points is taken, and additional alignment points are added heuristically. For each phrase pair, two phrasal translation probabilities, $P(\bar{f}|\bar{e})$ and $P(\bar{e}|\bar{f})$, are computed (one for each direction) from the relative frequency estimate on the training data, e.g.:

$$P(\bar{e}|\bar{f}) = \frac{count(\bar{e},\bar{f})}{count(\bar{f})} \quad (2)$$

Two analogous lexical scores are computed, e.g.:

$$Score_{lex}(\bar{f}|\bar{e}) = \prod_{j=1}^{J} \frac{1}{|\{i|a(i)=j\}|} \sum_{a(i)=j} p(f_j|e_i) \quad (3)$$

where j ranges over words in phrase $\bar{f}$ and i ranges over words in phrase $\bar{e}$. Here, we use two phrase tables concomitantly, one trained from each data source (BTEC and Europarl). We use the two phrase tables jointly, without renormalization of probabilities. An additional binary feature in the log-linear combination indicates which data source a given phrase pair comes from. However, the feature was shown to not have a significant impact on translation performance and is omitted for the second pass optimization.

4.2. Language Model

The first-pass language model is a trigram trained on the English side of the BTEC training set using modified Kneser-Ney smoothing. Further language models used during rescoring are described below.

5. Rescoring

The rescoring stage uses the first pass model scores along with five additional scores, as described below. Scores are again combined according to a log-linear model. Combination weights are trained to maximize the BLEU score on the development set using a downhill simplex search (i.e. amoeba search) [5]. The five additional scores are:

- a 4-gram language model score
- a POS n-gram score
- rank in N-best list
- Factored Language Model score ratio, and
- focused language model score

The last three are novel features in our system.

4-gram language model score (lm)
This is the score of a 4-gram language model trained on the English side of the BTEC training corpus using modified Kneser-Ney smoothing.

POS n-gram model score (pos)
The part-of-speech (POS) sequence of a given target sentence can be indicative of the sentence's syntactical well-formedness and thus translation fluency. Although it was cautioned in [6] that applying POS taggers directly to MT hypotheses may generate unexpected results (e.g. inserting a verb tag when there is no verb in the sentence), in practice we have found it useful to apply a POS language model to our N-best lists. We obtain POS annotations by applying the Maximum Entropy tagger of [7]. This tagger has been trained on the Wall Street Journal corpus; we apply it directly to our training set and N-best lists. In order to increase the training data for the POS n-gram we also used the Fisher

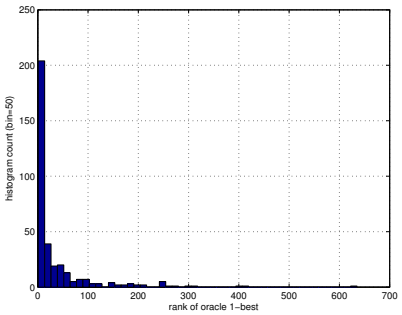

Figure 1: Histogram of oracle 1-best ranks (correct transcription condition).

corpus. Despite its different domain, this corpus also conversational in style and POS-level information may be transferable. We trained separate 5-gram POS language models on the training set and the Fisher corpus and combined them via interpolation.

Rank in N-best list (rank)
The first-pass decoder already generates high-quality N-best lists, in which the oracle-best hypotheses are typically ranked near the top of the list (as can be seen from the histogram in Figure 1). Ideally, the second pass should be guided by the ranking produced by the first-pass system. Moreover, the N-best lists contain many duplicate hypotheses at different positions, since the same sentence can be generated by many different phrase segmentations. Knowledge of which hypotheses are identical in terms of their word sequence should be utilized for rescoring. We therefore use a rank feature that (a) indicates the rank of a hypothesis in the first-pass N-best list, and (b) ties together identical hypotheses. The value of this feature is equivalent to the position of the hypothesis in the N-best list unless an identical, higher-ranked hypothesis has already been found. In that case, it takes on the value of the higher-ranked hypothesis. An example N-best list with ranks for each hypothesis is as follows:

1. the store is open on sundays (rank:1)
2. the store is open on sundays (rank:1)
3. the shop is open on sundays (rank:2)
4. the store is open on sundays (rank:1)
5. the store is it open on sundays (rank:3)

For our experiments, we slightly modified the above rank feature by applying a log function to the raw values. This bounds the features to a smaller range, similar to that of other features in the log-linear combination. We found that this did slightly better in our experiments than raw integer ranks. As shown in experiments (Section 8), the rank feature, is consistently the most useful feature in rescoring despite its simplicity.

Ratio of Factored Language Model scores (ratio)
Factored Language Models (FLMs) [8] are a flexible language modeling framework that can incorporate diverse sources of information, such as morphology, POS tags, etc. Previous experiments on using FLMs to rescore machine translation N-best lists have seen mixed results: little gain was shown for translation into English [9] but larger gains were shown for translation into Spanish, a morphologically richer language, especially under mismatched conditions [10]. Here, we use FLMs with three sources of information: words, part-of-speech, and data-driven word clusters in a trigram context. Word clusters were obtained by Brown clustering [11] using 500 word classes.

In this work, we apply FLMs to rescore English, but improve upon previous attempts by using two FLMs together in a discriminative fashion. In order to train the backoff structure and smoothing options of an FLM we use a genetic algorithm [12]. This requires a held-out set for iteratively optimizing the model parameters. While normally the references for some development set would be used for this purpose, in the context of machine translation we use the oracle-best hypotheses from the first pass, to ensure that the model is optimized on hypotheses that are likely to result in a good BLEU score. Here, we form two held-out sets, one consisting of the set of oracle best hypotheses from the N-best lists, and the other consisting of the set of oracle *worst* hypotheses from the N-best lists. The FLM optimized on the oracle best hypotheses should give high probability to sentences with *high* BLEU scores, while the FLM optimized on the oracle worst sentences will give high probability to sentences with *low* BLEU scores. The score used for rescoring then is $\phi_{ratio}(e) = \frac{FLM_1(e)}{FLM_2(e)}$, where $FLM_i(e),\ i = 1, 2$ is the probability of sentence e evaluated by the first and second FLMs, respectively. This method is analogous to the "splitting" technique used in [13], which divides the N-best list into good and bad sentences for training a perceptron-style learner. Our method differs in that instead of using a discriminative classifier, we use two generative models (FLMs) and take the log-probability ratio. This allows us to take advantage of the estimation techniques developed for language models.

Focused LM (focus, focusF)
The focused language model is a dynamically generated language model that focuses only on those words that occur in the N-best list. During the training phase of rescoring, we collect all the words in the N-best lists and use our training data to estimate a 4-gram model restricted to this vocabulary. This is done in order to force the language model to better discriminate between those n-grams that actually occur in the first-pass N-best lists. During the testing phase of rescoring, we again collect all words in the (test set) N-best list and estimate another focused LM. The score in both cases is the log-probability of each hypothesis assigned by the respective focused LMs. Note that the weight for this score is optimized jointly with all other scores on the development set and is held fixed for the test set, although the language

model itself changes. This may be a potential weakness; further work remains on analyzing the extent to which the language model and vocabulary differences affect rescoring. We used two kinds of focused LM features in our experiments. ϕ_{focus} uses LMs trained on the BTEC training set only, while ϕ_{focusF} includes Fisher data as well.

6. Postprocessing

For postprocessing we use a hidden-event n-gram model [14, 15] to restore punctuation and a noisy-channel model for truecasing. The hidden-ngram model partitions the vocabulary or event set E into two (possibly overlapping) subsets: the set W of regular words and the set H of "hidden events", in this case the set of punctuation signs. During training, all events are observed; thus, training a model that predicts the joint probability of hidden and observed words is equivalent to training a standard n-gram model on punctuated text:

$$P(e_1, ..., e_T) \approx \prod_{t=n}^{T} P(e_t | e_{t-1}, ..., e_{t-n+1}) \qquad (4)$$

During testing, hidden events are hypothesized after every word. Their posterior probability is computed by using a forward-backward dynamic programming procedure and the transition probabilities provided by the trained n-gram model. The noisy-channel model consists of a 4-gram model trained over a mixed-case representation of the BTEC training corpus and a probabilistic mapping table for lowercase-uppercase word variants. It was implemented using the *disambig* tool from the SRILM package [16].

Finally, we also morphologically decompose and translate unknown words, similar to the procedure described in [10]. In the case of Italian, this means that cliticized pronouns are detached from the end of the word before translation.

7. Spoken-Language Specific Processing

In order to take advantage of the additional information available for the ASR-output condition, we attempted to use the ASR N-best lists provided (N = 20). We translated all N-best hypotheses directly (producing M translation hypotheses per input sentence) and optimized our system using the entire set of $N \times M$-best translations. Table 1 shows a comparison of the BLEU and PER (position-independent word error rate) scores obtained by the oracle hypotheses from both types of input. As can be seen, the BLEU score improves by 2.7% absolute and the PER decreases by 2.1%. However, in initial rescoring experiments we did not obtain an improvement from the $N \times M$ list, so that they were not used for the evaluation system.

However, further post-evaluation experiments using N-best lists are described below.

	BLEU (%)	PER
1-best	44.1	34.7
N-best	46.8	32.8

Table 1: Oracle translation scores on 1-best vs. N-best ASR hypotheses

8. Experiments

We first investigated how the coverage of phrases up to length 7 was changed by adding the Europarl data. Table 2 shows the percentages of phrases in the development and test sets for which a match can be found in the phrase tables trained from a the different corpora. As can be seen, the coverage of short phrases up to 3 words improves noticeably while the coverage of longer phrases hardly changes. However, improved coverage does not necessarily result in better translations since the translations for the newly covered phrases may not match the references. The effect on actual translation performance on the development data is shown in Table 3. Both BLEU and PER are improved.

	BTEC	Europarl	combined
1	84.0 (76.6)	88.3 (85.9)	94.0 (91.6)
2	40.8 (37.7)	48.1 (46.2)	60.1 (56.9)
3	13.6 (12.9)	11.9 (12.6)	20.1 (20.6)
4	3.4 (4.0)	1.5 (1.8)	4.5 (5.4)
5	1.1 (0.9)	0.2 (0.2)	1.3 (1.1)
6	0.3 (0.2)	0.0 (0.0)	0.3 (0.2)
7	0.1 (0.0)	0.0 (0.0)	0.1 (0.0)

Table 2: Coverage of phrases (in %) in the development set (test set) for individual and combined training corpora (correct transcription condition).

	BLEU (%)	PER
BTEC only	44.5	29.9
+ Europarl	**46.8**	**28.0**

Table 3: First-pass translation performance with BTEC training data alone and with added Europarl data on the development set (correct transcription condition).

8.1. Rescoring results

In the rescoring experiments we first tested the impact of the added Fisher data for various language models. Table 4 shows that the Fisher data only gave a slight improvement in BLEU score for the focused language model in the correct transcription condition, PER scores did not change.

For finding the best combination of second-pass rescoring features we begin by using the 9 features of the first-pass

Features used in rescoring	BLEU	PER
base+lm w/o Fisher	44.9	31.0
base+lm w/ Fisher	44.8	31.0
base+pos w/o Fisher	45.8	30.8
base+pos w/ Fisher	45.9	30.8
base+focus (w/o Fisher)	44.4	31.3
base+focusF (w/ Fisher)	45.1	31.6

Table 4: Effect of added Fisher data on language model training. Scores shown are second-pass scores on the dev set (correct transcription condition) with baseline features and one added language model feature.

system and iteratively add new features in a greedy fashion. Tables 5 and 6 show the BLEU/PER scores of the correct transcription and ASR-output conditions, respectively. In both cases, the best individual feature among the five discussed above is rank, which yields noticeable improvements of 1-2% absolute in BLEU and 1.8-2.3% PER. All other rescoring features present mixed results when used in isolation together with the first-pass features; some improve BLEU but not PER. There also seem to be significant interactions between the individual features; the best combination of all 14 features improve both BLEU and PER significantly compared to only using the first-pass features. A comparison with the oracle scores also shows that there is still room for improvement in the second-pass rescoring.

Features used in rescoring	K	BLEU	PER
baseline features	9	44.8[1]	30.8
base+focus	10	44.4	31.3
base+lm	10	44.9	31.0
base+ratio	10	45.0	31.4
base+focusF	10	45.1	31.6
base+pos	10	45.9	30.8
base+rank	10	46.8	28.5
base+rank+focusF	11	47.5	28.1
base+rank+focusF+pos	12	47.4	29.1
base+rank+focusF+pos+ratio	13	47.2	29.3
*base+rank+focusF+pos+ratio+lm	14	**47.6**	**28.0**
Comparison systems	K	BLEU	PER
First-pass decoding	n/a	46.8	28.0
Oracle 1-best in N-best list	n/a	59.7	21.4

Table 5: BLEU and PER scores (%) for correct transcription translation results (on the development set). K is the number of feature used in rescoring. *The rescorer used in official evaluation.

[1]The baseline rescorer starts out at a worse performance level than 1-st pass decoding (e.g., 44.8 vs. 46.8 BLEU). This is because phrase table scores are used differently in decoding vs. rescoring by the Pharaoh decoder when multiple entries per phrase pair are present.

Features used in rescoring	K	BLEU	PER
baseline features	9	34.6	39.6
base+focusF	10	34.3	40.9
base+lm	10	34.3	39.2
base+ratio	10	34.4	39.4
base+focus	10	35.2	39.8
base+rank	10	35.6	37.8
base+pos	10	35.7	40.0
base+pos+rank	11	36.1	37.4
base+pos+rank+ratio	12	36.6	37.6
base+pos+rank+ratio+focus	13	36.9	**37.3**
*base+pos+rank+ratio+focus+lm	14	**37.0**	37.8
Comparison systems	K	BLEU	PER
1st-pass decoding	n/a	35.4	37.3
oracle 1-best in N-best list	n/a	46.5	31.2

Table 6: BLEU and PER scores (%) for ASR-output translation (on the development set). *The rescorer used in official evaluation.

9. Results

The official evaluation results are shown in Tables 7 and 8. Table 8 shows that our system almost always ranks highest when evaluated on lowercase text and without punctuation. However, when evaluated on true-case and with punctuation, the systems shows stronger relative degradation than other systems, suggesting that its post-processing component requires further work. As expected, translation from ASR-output is significantly worse than translation from the correct transcription, about 8 percentage points absolute in both BLEU and PER.

	BLEU	PER	WER	NIST	METEOR
	Correct Transcription				
case/punc	35.43	38.92	48.34	8.19	70.17
without	42.06	31.75	42.86	9.24	70.19
	ASR-Output				
case/punc	27.87	46.76	55.87	6.93	58.53
without	31.68	42.11	53.17	7.69	58.53

Table 7: Official evaluation results. case/punc = with case and punctuation taken into account, without = without case or punctuation.

10. Post-Evaluation Analyses

10.1. Data

With the exception of separate weight optimization runs, the system used for the ASR-output condition was identical to that used for the correct transcription condition. After the official evaluation we conducted additional experiments to further optimize the ASR-output system. First, we looked at the impact of adding data separately for this system. Tables 9

	BLEU	PER	WER	NIST	METEOR
	Correct Transcription				
case/punc	3	1	3	1	1
without	1	1	1	1	1
	ASR-Output				
case/punc	5	4	4	1	1
without	2	1	1	1	1

Table 8: Rank out of 11 submissions, according to official evaluation results.

and 10 are analogous to Tables 2 and 3 above and show that, contrary to the correct transcription condition, the Europarl data only helps the PER score slightly but actually degrades BLEU. A similar analysis for the added language model data also shows a different pattern than for the correct transcription condition: only the POS n-gram was improved slightly by the Fisher data, the other language models deteriorated (see Table 11).

	BTEC	Europarl	combined
1	84.0 (81.0)	87.7 (88.1)	94.6 (94.9)
2	38.9 (36.2)	43.0 (41.9)	54.7 (52.4)
3	13.6 (12.3)	9.9 (10.4)	19.1 (18.2)
4	4.2 (3.6)	1.0 (1.3)	4.9 (4.6)
5	1.4 (0.9)	0.2 (0.2)	1.6 (1.0)
6	0.4 (0.1)	0.0 (0.0)	0.4 (0.1)
7	0.2 (0.0)	0.1 (0.0)	0.2 (0.0)

Table 9: Coverage of phrases (in %) in the development set (test set) for individual and combined training corpora (1-best ASR output).

	BLEU (%)	PER
BTEC only	**36.5**	38.0
+ Europarl	35.4	**37.3**

Table 10: First-pass translation performance with BTEC training data alone and with added Europarl data on the development set - 1-best ASR output.

10.2. Confusion Networks

As an alternative to direct N-best translation we investigated the use of confusion network representations to transform the ASR-output hypotheses. Confusion networks [17] are a compact representation of multiple sentence hypotheses derived from a word lattice or an N-best list. They take the form of a connected graph with a designated start and end node, where edges between nodes represent different competing word hypothesis for a given position in the sentence (see Figure 2). In addition, edges associated with posterior probabilities of

Features used in rescoring	BLEU	PER
base+lm w/o Fisher	34.3	39.2
base+lm w/ Fisher	34.1	39.6
base+pos w/o Fisher	35.4	40.2
base+pos w/ Fisher	35.7	40.0
base+focus (w/o Fisher)	35.2	39.8
base+focusF (w/ Fisher)	34.3	40.9

Table 11: Effect of added Fisher data on language model training. Scores shown are second-pass scores on the dev set (ASR-output) with baseline features and one added language model feature.

	BLEU (%)	PER
1-best from recognizer	36.5	38.0
1-best from confusion net	37.4	37.7
+ second-pass rescoring	**38.0**	**36.8**

Table 12: Translation performance based on 1-best ASR-output vs. 1-best hypothesis selected from confusion network representation of recognizer N-best list (dev set).

the word hypotheses. In order to construct a confusion network from an N-best list, all N-best hypotheses are aligned into a grid of word positions defined by the first hypothesis. The posterior probabilities are then obtained by the frequency count of a given word label in that position relative to the total count of words in that position.

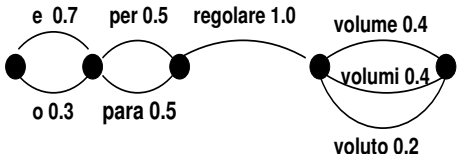

Figure 2: Confusion network.

Confusion networks have been used in machine translation in order to e.g. combine the output from multiple translation systems [18] or to perform translation directly from a confusion network instead of a word lattice or an N-best list [19]. Here, we use a confusion network to obtain better ASR input hypotheses; however, translation is still done from a single hypothesis per sentence. Having obtained the posterior probabilities, we construct a new 1-best hypothesis by choosing the highest-probability word at each position. This may result in more reliable hypotheses as well as hypotheses that were not in the original N-best list. The translation results on the development set (Table 12) show an improvement compared to the previous system.

Translation was then rerun on the test set with the improved data selection and confusion network based selection of input hypotheses. However, results on the test set did not improve compared to the official evaluation results. Further

analysis of this is being performed.

11. Conclusions

We have presented a multi-pass phrase-based SMT system for the Italian-English BTEC translation task. Our focus was on adding out-of-domain data to both the translation and the language model, novel features for rescoring, and using N-best information for ASR-output translation. Our conclusions are:

1. Adding data from different domains and styles is of mixed benefit. Data from parliamentary proceedings mostly helped the translation model for the correct transcription input condition. Additional English conversational data of a general nature did not for the most part improve the various target language models.

2. Of the several new features used during rescoring (factored language model score ratio, focused LM, rank in N-best list), several features showed small gains, especially in combination. The rank feature clearly yields a significant improvement by itself.

3. With respect to using N-best information for ASR-output translation, we found that direct translation of N-best lists was not useful. Confusion network based selection of 1-best input hypotheses was helpful on the development data but did not yet show any improvement on the test set.

Acknowledgements
This work was supported by grant IIS-0308297 from the U.S. National Science Foundation and by an National Science Foundation graduate research fellowship for the second author. We would also like to thank Mei Yang for help with the post-evaluation analysis.

12. References

[1] Koehn, P., "Europarl: A Multilingual Corpus for Evaluation of Machine Translation", Unpublished Manuscript

[2] Koehn, P. "Pharaoh: a beam search decoder for phrase-based statistical machine translation models", Proceedings of AMTA (Assoc. for Machine Translation of the Americas), 2004

[3] Och, F.J., "Minimum Error Rate Training for Statistical Machine Translation", in Proc. of 41st Meeting of the Association for Computational Linguistics, 2003.

[4] Och, F.J., and Ney, H., "A systematic comparison of various statistical alignment models", Computational Linguistics 29(1), 19-52, 2003

[5] Nelder, J. A. and Mead, R. "A simplex method for function minimization", Computing Journal, 7(4):308–313, 1965.

[6] Och, F.J., et. al., "A smorgasbord of features for statistical machine translation", in Proc. of Human Language Technology (HLT/NAACL), 2004.

[7] Ratnaparkhi, A., "A maximum entropy part-of-speech tagger", in Proc. of Empirical Methods in Natural Language Processing (EMNLP), 1996.

[8] Bilmes, J. and Kirchhoff, K., "Factored language models and generalized parallel backoff", in Proc. of Human Language Technology Conference (HLT/NAACL), 2003

[9] Kirchhoff, K., Yang, M., "Improved language modeling for statistical machine translation", in Proc. of ACL Workshop on Building and Using Parallel Texts, 2005

[10] Kirchhoff, K., Yang, M., Duh, K., "Statistical machine translation of parliamentary proceedings using morpho-syntactic knowledge", TC-Star Speech to Speech Translation Workshop, 2006.

[11] Brown, P., Della Pietra, V.J., deSouza, P.V., Lai, J.C., and Mercer, R.L., "Class-based n-gram models of natural language", Computational Linguistics 18(4),1992, 467-479

[12] Duh, K. and Kirchhoff, K., "Automatic Learning of Language Model Structure", in Proc. of 20th Int'l Conf. on Computational Linguistics (COLING), 2004.

[13] Shen, L., Sarkar, A., Och, F.J., "Discriminative Reranking for Machine Translation", Proc. of Human Language Technology (HLT/NAACL), 2004.

[14] Stolcke, A. and Shriberg, E., "Statistical language modeling for speech disfluencies", Proc. of Int'l Conference on Acoustics, Speech, and Signal Processing (ICASSP), 1996, 405-409

[15] Stolcke, A. and Shriberg, E., "Automatic linguistic segmentation of conversational speech", Proc. of Int'l Conf. on Spoken Language Processing (ICSLP), 1996, 1005-1008

[16] Stolcke, A., "SRILM - an extensible language modeling toolkit", Proc. of Int'l Conf. on Spoken Language Processing (ICSLP), 2002,901-904

[17] Mangu, L., Brill, E., and Stolcke, A. "Finding Consensus Among Words: Lattice-based Word Error Minimization.", Proceedings of Eurospeech, vol. 1, 495-498, 1999

[18] Matusov, E., Ueffing, N. and Ney, H. “Computing consensus translation from multiple machine translation systems using enhanced hypotheses alignment”, Proceedings of EACL (European Assoc. of Computational Linguistics), 2006

[19] Bertoldi, N. and Federico, M., “A new decoder for spoken language translation based on confusion networks”, Proc. of the IEEE Automatic Speech Recognition and Understanding Workshop, 2005