The RWTH Statistical Machine Translation System for the IWSLT 2006 Evaluation

Arne Mauser, Richard Zens, Evgeny Matusov, Saša Hasan, Hermann Ney

Human Language Technology and Pattern Recognition
Lehrstuhl für Informatik 6, Computer Science Department
RWTH Aachen University, D-52056 Aachen, Germany
{mauser,zens,matusov,hasan,ney}@cs.rwth-aachen.de

Abstract

We give an overview of the RWTH phrase-based statistical machine translation system that was used in the evaluation campaign of the International Workshop on Spoken Language Translation (IWSLT) 2006. The system was ranked first with respect to the BLEU measure in all language pairs it was used

Using a two-pass aproach, we first generate the N best translation candidates. The second pass consists of rescoring and reranking these candidates. We will give a description of the search algorithm as well as of the models used in each pass.

We will also describe our method for dealing with punctuation restoration, in order to overcome the difficulties of spoken language translation.

This work also includes a brief description of the system combination done by the partners participating in the European TC-Star project.

1. Introduction

We give an overview of the RWTH phrase-based statistical machine translation system that was used in the evaluation campaign of the International Workshop on Spoken Language Translation (IWSLT) 2006.

We use a two pass approach. First, we generate a list of the N best translation candidates. Then, we apply additional models in a rescoring/reranking approach.

This work is structured as follows: first, we will review the statistical approach to machine translation and introduce the notation that we will use in the later sections. Then, we will describe the models and algorithms that are used for generating the N-best list, i.e., the first pass. In Section 3, we will describe the models that are used to rescore and rerank this N-best list, i.e., the second pass. Afterwards, we will give an overview of the tasks and discuss the experimental results. This paper will also include a section describing the method used for the system combination of the TC-Star project partners.

The overall system is similar to the one used in the 2005 IWSLT evaluation [1]. However, it contains novel features for the first pass, as well as for the second pass. In the first pass, we use phrase count features (cf. 2.2) to smooth the phrase probabiliies. In the second pass, we used sentence mixture language models 3.2 as a new model for rescoring.

1.1. Source-channel approach to SMT

In statistical machine translation, we are given a source language sentence
$f_1^J = f_1 \dots f_j \dots f_J$, which is to be translated into a target language sentence $e_1^I = e_1 \dots e_i \dots e_I$.

Among all possible target language sentences, we will choose the sentence with the highest probability:

$$\hat{e}_1^{\hat{I}} = \operatorname*{argmax}_{I, e_1^I} \left\{ Pr(e_1^I | f_1^J) \right\} \quad (1)$$

$$= \operatorname*{argmax}_{I, e_1^I} \left\{ Pr(e_1^I) \cdot Pr(f_1^J | e_1^I) \right\} \quad (2)$$

This decomposition into two knowledge sources is known as the source-channel approach to statistical machine translation [2]. It allows for an independent modeling of the target language model $Pr(e_1^I)$ and the translation model $Pr(f_1^J|e_1^I)$[1].

The target language model describes the well-formedness of the target language sentence. The translation model links the source language sentence to the target language sentence. The argmax operation denotes the search problem, i.e., the generation of the output sentence in the target language.

1.2. Log-linear model

A generalization of the classical source-channel approach is the direct modeling of the posterior probability $Pr(e_1^I|f_1^J)$. Using a log-linear model [3], we obtain:

$$Pr(e_1^I | f_1^J) = \frac{\exp\left(\sum_{m=1}^M \lambda_m h_m(e_1^I, f_1^J)\right)}{\sum_{e'^{I'}_1} \exp\left(\sum_{m=1}^M \lambda_m h_m(e'^{I'}_1, f_1^J)\right)} \quad (3)$$

[1] The notational convention will be as follows: we use the symbol $Pr(\cdot)$ to denote general probability distributions with (nearly) no specific assumptions. In contrast, for model-based probability distributions, we use the generic symbol $p(\cdot)$.

The denominator represents a normalization factor that depends only on the source sentence f_1^J. Therefore, we can omit it during the search process. As a decision rule, we obtain:

$$\hat{e}_1^{\hat{I}} = \underset{I,e_1^I}{\operatorname{argmax}} \left\{ \sum_{m=1}^{M} \lambda_m h_m(e_1^I, f_1^J) \right\} \quad (4)$$

This is a generalization of the source-channel approach. It has the advantage that additional models $h(\cdot)$ can be easily integrated into the overall system. The model scaling factors λ_1^M are trained according to the maximum entropy principle, e.g., using the GIS algorithm. Alternatively, one can train them with respect to the final translation quality measured by an error criterion [4]. For the IWSLT evaluation campaign, we optimized the scaling factors with respect to the BLEU measure, using the Downhill Simplex algorithm from [5].

1.3. Phrase-based approach

The basic idea of phrase-based translation is to segment the given source sentence into phrases, then translate each phrase and finally compose the target sentence from these phrase translations. This idea is illustrated in Figure 1. Formally, we define a segmentation of a given sentence pair (f_1^J, e_1^I) into K blocks:

$$k \rightarrow s_k := (i_k; b_k, j_k), \text{ for } k = 1 \ldots K. \quad (5)$$

Here, i_k denotes the last position of the k^{th} target phrase; we set $i_0 := 0$. The pair (b_k, j_k) denotes the start and end positions of the source phrase that is aligned to the k^{th} target phrase; we set $j_0 := 0$. Phrases are defined as nonempty contiguous sequences of words. We constrain the segmentations so that all words in the source and the target sentence are covered by exactly one phrase. Thus, there are no gaps and there is no overlap.

For a given sentence pair (f_1^J, e_1^I) and a given segmentation s_1^K, we define the bilingual phrases as:

$$\tilde{e}_k := e_{i_{k-1}+1} \ldots e_{i_k} \quad (6)$$

$$\tilde{f}_k := f_{b_k} \ldots f_{j_k} \quad (7)$$

Note that the segmentation s_1^K contains the information on the phrase-level reordering. The segmentation s_1^K is introduced as a hidden variable in the translation model. Therefore, it would be theoretically correct to sum over all possible segmentations. In practice, we use the maximum approximation for this sum. As a result, the models $h(\cdot)$ depend not only on the sentence pair (f_1^J, e_1^I), but also on the segmentation s_1^K, i.e., we have models $h(f_1^J, e_1^I, s_1^K)$.

1.4. Source cardinality synchronous search

For single-word based models, this search strategy is described in [6]. The idea is that the search proceeds synchronously with the cardinality of the already translated source positions. Here, we use a phrase-based version of this idea. To make the search problem feasible, the reorderings are constrained as in [7].

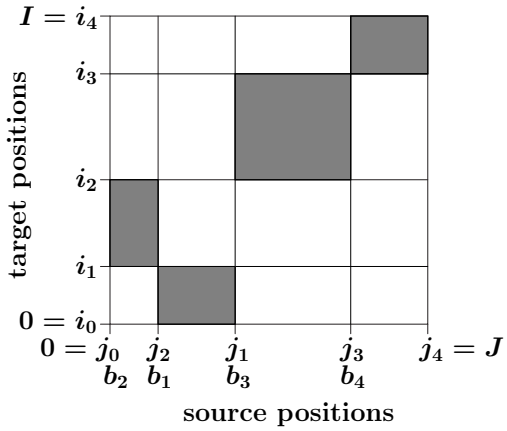

Figure 1: Illustration of the phrase segmentation.

2. Models used during search

When searching for the best translation for a given input sentence, we use a log-linear combination of several models (also called feature functions) as decision criterion. In this section, we will describe the models that are used in the first pass, i.e., during N best list generation. More specifically the models are: a phrase translation model, a word-based translation model, word and phrase penalty, a target language model and a reordering model. We will now describe the models in detail.

2.1. Phrase-based model

The phrase-based translation model is the main component of our translation system. The hypotheses are generated by concatenating target language phrases. The pairs of source and corresponding target phrases are extracted from the word-aligned bilingual training corpus by the phrase extraction algorithm described in detail in [8]. The main idea is to extract phrase pairs that are consistent with the word alignment, meaning that the words of the source phrase are aligned only to words in the target phrase and vice versa. This criterion is identical to the alignment template criterion described in [9].

We use relative frequencies to estimate the phrase translation probabilities:

$$p(\tilde{f}|\tilde{e}) = \frac{N(\tilde{f}, \tilde{e})}{N(\tilde{e})} \quad (8)$$

Here, the number of co-occurrences of a phrase pair $(\tilde{f}, \tilde{e})$ that are consistent with the word alignment is denoted as $N(\tilde{f}, \tilde{e})$. If one occurrence of a target phrase $\tilde{e}$ has $N > 1$ possible translations, each of them contributes to $N(\tilde{f}, \tilde{e})$

with $1/N$. The marginal count $N(\tilde{e})$ is the number of occurrences of the target phrase $\tilde{e}$ in the training corpus. The resulting feature function is:

$$h_{\text{Phr}}(f_1^J, e_1^I, s_1^K) = \log \prod_{k=1}^{K} p(\tilde{f}_k|\tilde{e}_k) \quad (9)$$

To obtain a more symmetric model, we use the phrase-based model in both directions $p(\tilde{f}|\tilde{e})$ and $p(\tilde{e}|\tilde{f})$.

2.2. Phrase Count Features

The reliability of the phrase probability estimation is largely dependent on the amount and quality of the training data. Generally, the probability of rare phrases tends to be overestimated, but as they do not occur often, it might be as well errors originating from mistranslations in the training data or erroneous word alignments. Therefore, we also included features based on the actual count of the bilingual phrase pair.

$$h_{\text{C},\tau}(f_1^J, e_1^I, s_1^K) = \sum_{k=1}^{K} [N(\tilde{f}_k, \tilde{e}_k) \leq \tau]$$

We use $[\cdot]$ to denote a true or false statement [10], i.e., the result is 1 if the statement is true, and 0 otherwise. In general, we use the following convention:

$$[\mathcal{C}] = \begin{cases} 1, & \text{if condition } \mathcal{C} \text{ is true} \\ 0, & \text{if condition } \mathcal{C} \text{ is false} \end{cases} \quad (10)$$

The value τ determines the threshold for the phrase count feature. In the evaluation system, we used three phrase count features with τ manually chosen and ranging from 1.0 to 3.0. As that actual phrase count values are fractional, also fractional thresholds can be used.

2.3. Word-based lexicon model

We are using relative frequencies to estimate the phrase translation probabilities. Most of the longer phrases occur only once in the training corpus. Therefore, pure relative frequencies overestimate the probability of those phrases. To overcome this problem, we use a word-based lexicon model to smooth the phrase translation probabilities.

The score of a phrase pair is computed similar to the IBM model 1, but here, we are summing only within a phrase pair and not over the whole target language sentence:

$$h_{\text{Lex}}(f_1^J, e_1^I, s_1^K) = \log \prod_{k=1}^{K} \prod_{j=b_k}^{j_k} \sum_{i=i_{k-1}+1}^{i_k} p(f_j|e_i) \quad (11)$$

The word translation probabilities $p(f|e)$ are estimated as relative frequencies from the word-aligned training corpus. The word-based lexicon model is also used in both directions $p(f|e)$ and $p(e|f)$.

2.4. Word and phrase penalty model

In addition, we use two simple heuristics, namely word penalty and phrase penalty:

$$h_{\text{WP}}(f_1^J, e_1^I, s_1^K) = I \quad (12)$$

$$h_{\text{PP}}(f_1^J, e_1^I, s_1^K) = K \quad (13)$$

These two models affect the average sentence and phrase lengths. The model scaling factors can be adjusted to prefer longer sentences and longer phrases.

2.5. Target language model

We use the SRI language modeling toolkit [11] to train a standard n-gram language model. The resulting feature function is:

$$h_{\text{LM}}(f_1^J, e_1^I, s_1^K) = \log \prod_{i=1}^{I} p(e_i|e_{i-n+1}^{i-1}) \quad (14)$$

The smoothing technique we apply is the modified Kneser-Ney discounting with interpolation. We used a 6-gram language model for all tasks.

2.6. Reordering model

We use a very simple reordering model that is also used in, for instance, [9, 12]. It assigns costs based on the jump width:

$$h_{\text{RM}}(f_1^J, e_1^I, s_1^K) = \sum_{k=1}^{K} |b_k - j_{k-1} - 1| + J - j_K \quad (15)$$

3. Rescoring models

In this section, we describe the second pass of our system, the rescoring of N-best lists. N-best lists are suitable for easily applying several rescoring techniques because the hypotheses are already fully generated. In comparison, word graph rescoring techniques need specialized tools which traverse the graph appropriately. Additionally, because a node within a word graph allows for many histories, one can only apply local rescoring techniques, whereas for N-best lists, techniques can be used that consider properties of the whole target sentence.

In the next sections, we will present several rescoring models.

3.1. Clustered language models

One of the first ideas in rescoring is to use additional language models that were not used in the generation procedure. In our system, we use clustered language models based on regular expressions [13]. Each hypothesis is classified by matching it to regular expressions that identify the type of the sentence. Then, a cluster-specific (or sentence-type-specific) language model is interpolated into a global language model

		Chinese	Japanese	English
Train:	Sentences	40 000		
	Running Words	295 579	348 103	377 355
	Vocabulary	11 170	12 533	9 570
	Singletons	4 348	5 572	3 904
IWSLT'05	Sentences	506		
	Running Words	3 208	3 601	3 767
	Vocabulary	928	950	843
	OOVs (running words)	67 (2.1%)	46 (1.3%)	179 (4.7%)
DEV'06	Sentences	489		
	Running Words	5 214	5 874	6 362
	Vocabulary	1 137	1 189	1 012
	OOVs (running words)	126 (2.4%)	119 (2.0%)	296 (4.7%)
EVAL'06	Sentences	500		
	Running Words	5 550	6 489	
	Vocabulary	1 328	1 330	
	OOVs (running words)	172 (3.1%)	170 (2.6%)	

Table 1: Corpus Statistics of the IWSLT 2006 training data and development, test and eval corpora after preprocessing

to compute the score of the sentence:

$$h_{\text{CLM}}(f_1^J, e_1^I) = \log \sum_c \left[\mathcal{R}_c(e_1^I)\right] \left(\alpha_c p_c(e_1^I) + (1-\alpha_c) p_g(e_1^I)\right), \quad (16)$$

where $p_g(e_1^I)$ is the global language model, $p_c(e_1^I)$ the cluster-specific language model, and $\left[\mathcal{R}_c(e_1^I)\right]$ denotes the true-or-false statement (cf. Equation 10) which is 1 if the c^{th} regular expression $\mathcal{R}_c(\cdot)$ matches the target sentence e_1^I and 0 otherwise.[2] Typical examples for clusters are questions and exclamations, which can usually be detected by punctuation marks and/or specific words (i.e. "what", "when", "how", ... at the beginning of a question sentence. Furthermore, when looking at the training data, specific sentences and expressions can be spotted occur quite frequently and can be joined into a cluster.

3.2. Sentence-level Mixtures

As an additional language model in rescoring, we use sentence level mixture language models, as presented in [14]. The goal is to represent topic dependencies combining M different language models with a global one, corresponding to the index $m = 0$ in the following equation (for the case of trigram language models)

$$p(e_1^I) = \sum_{m=0}^{M} \lambda_m \left[\prod_{i=1}^{I} p_m(e_i|e_{i-1}, e_{i-2})\right]. \quad (17)$$

The training sentences are automatically divided into a fixed number M of clusters (representing different topics) using a maximum likelihood approach and the weights λ_m are trained on the development data. We used 5-grams for this rescoring model.

[2]The clusters are disjunct, thus only one regular expression matches.

3.3. IBM model 1

IBM model 1 rescoring rates the quality of a sentence by using the probabilities of one of the easiest single-word based translation models:

$$h_{\text{IBM1}}(f_1^J, e_1^I) = \log \left(\frac{1}{(I+1)^J} \prod_{j=1}^{J} \sum_{i=0}^{I} p(f_j|e_i)\right) \quad (18)$$

Despite its simplicity, this model achieves good improvements [15].

3.4. IBM1 deletion model

During the IBM model 1 rescoring step, we make use of another rescoring technique that benefits from the IBM model 1 lexical probabilities:

$$h_{\text{Del}}(f_1^J, e_1^I) = \sum_{j=1}^{J} \prod_{i=0}^{I} \left[p(f_j|e_i) < \tau\right] \quad (19)$$

We call this the IBM1 deletion model. It counts all source words whose lexical probability given each target word is below a threshold τ. In the experiments, τ was chosen between 10^{-1} and 10^{-4}.

3.5. Sentence length model

Sentence length is crucial for the evaluation of machine translation output, especially when using automatic evalua-

tion measures. Therefore we explicitly modeled the target sentence length I using the method described in [16]:

$$h_{\mathrm{SL}}(f_1^J, e_1^I) = \log \sum_{e_1^I} p(e_1^I | f_1^J)$$

The sum is carried out only over those target hypotheses that have length I.

4. Tasks and corpora

The experiments were carried out on the *Basic Travel Expression Corpus* (BTEC) task [17]. 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 1. For the open data track a 40 000 sentences training corpus and four test sets were made available for each language pair. Other resources, despite proprietary data were permitted, but were not used in this system.

As the BTEC is a rather clean corpus, the preprocessing consisted mainly of tokenization, i.e., separating punctuation marks from words. Additionally, we expanded contractions such as *it's* or *I'm* in the English corpus and we removed the case information. There was no special preprocessing for the Chinese and the Japanese training corpora.

We used the provided IWSLT 2006 development set to optimize the system, for instance, the model scaling factors and the GIZA++ [18] parameter settings. The IWSLT'05 test set was used as a blind test corpus. After the optimization, we added the development sets to the training corpus and retrained the whole system.

5. Experimental results

The automatic evaluation criteria are computed using the IWSLT 2006 evaluation server. For all the experiments, we report the two accuracy measures BLEU [19] and NIST [20] as well as the two error rates WER and PER. For the primary submissions, we also report the two accuracy measure Meteor [21]. All those criteria are computed with respect to multiple references.

5.1. Primary submissions

The translation results of the RWTH primary submissions are summarized in Table 4.

5.2. Analysis of the results for text input

In Table 2, we show the progress of the RWTH machine translation over the past two years. The evaluation is done on the IWSLT 2005 test set for the supplied data track. For the 2006 system, we provide two variants. First, a system, that is only trained on 20k sentence pairs, as the systems from 2004 and 2005. Second, a system, that uses the full 40k sentence pairs that were used in the 2006 evaluation system. This makes the 2006 system comparable to the previous systems and also shows the effect of the additional data.

Table 2: Progress over time: comparison of the RWTH systems of the years 2004 to 2006 for the supplied data track on the IWSLT 2005 test set.

Translation Direction	System	BLEU [%]	NIST	WER [%]	PER [%]
Chin.-Engl.	2004	40.4	8.59	52.4	42.2
	2005	46.3	8.73	47.4	39.7
	2006	48.8	8.56	47.3	39.2
	2006 (40k)	51.4	9.00	40.0	33.2
Jap.-Engl.	2004	44.8	9.41	50.0	37.7
	2005	49.8	9.52	46.5	36.8
	2006	56.5	8.72	41.9	32.8
	2006 (40k)	57.1	8.69	41.8	33.6

Table 3: Rescoring: effect of successively adding models for the Chinese-English IWSLT 2006 development set.

System	BLEU [%]	NIST	WER [%]	PER [%]
Baseline	21.9	6.31	66.4	50.8
+CLM	22.5	6.09	63.7	49.7
+Len	23.0	6.36	66.7	51.3
+MIX	23.2	6.30	65.6	50.4
+Del	23.4	6.37	66.1	50.4
+IBM1	23.5	6.33	64.8	49.4

Even without the additional data, the systems improve in all scores except the NIST measure. Interestingly, using the double ammount of training data only slightly improves translation quality. This can be attributed to the fact, that the coverage of the IWSLT '04 test data is already high for the 20k sentences and the 16 references allow for a large tolerance in the MT output.

The effects of the N-best list rescoring for the IWSLT 2006 development set are summarized in Table 3. Improvements on the developmenet set were also verified on the IWSLT '04 to avoid overfitting.

6. Punctuation prediction and case restoration

When translating speech, the input of the translation system usually does not contain punctuation marks or case information. The human user of an MT system however expects readable output in the target language, including proper punctuation and capitalization. The IWSLT 2006 evaluation reflects these conditions. The input to the translation system was provided without punctuation and punctuation had to be generated by the MT system.

Table 4: Official results for the RWTH primary submissions on the IWSLT 2006 test set.

Translation Direction	Input	Accuracy Measures			Error Rates	
		BLEU [%]	NIST	Meteor [%]	WER [%]	PER [%]
Chinese-English	Correct	24.2	6.10	50.3	66.7	50.9
	Read	21.1	5.40	44.3	69.5	55.3
	Spont	19.0	5.05	42.0	71.2	57.1
Japanese-English	Correct	23.7	5.92	48.9	68.5	51.5
	Read	21.4	5.65	45.7	70.7	53.8

6.1. Punctuation prediction

When predicting punctuation in speech translation, we can follow three different paths:

1. Predicting punctuation on *source side* i.e. in the ASR output. In general, this strategy has the advantage, that prosodic cues from speech recognition can be used to help punctuation prediction. Furthermore, the MT system trained for regular text translation can be used, as it expects punctuation in the input. A disadvantage is that falsely inserted punctuation marks can deteriorate translation performance, especially of phrase-based MT systems, as long phrases might not match anymore.

2. Predicting punctuation on the *target side* (i.e. the MT output). Usually punctuation rules differ between languages. This strategy has the advantage, that punctuation is learned on the target language and thus expected to better reflect the corresponding punctuation rules. The disadvantage is, that the decision about inserting punctuation is based on MT output. This output is likely to contain errors which might lower the quality of punctuation prediction.

3. Predicting punctuation *implicitly* during the translation process. The MT system is trained without punctuation on the source side and with punctuation on the target side. This way, both the translation and the target language model are used to predict punctuation. The advantage is that the full predictive power of the MT system is used not only for the generation of words, but also for the generation of punctuation. Optimization of the scaling factors of the model can easily be done with respect to references with punctuation. The disadvantage is, that a separate MT system needs to be trained for this condition.

We decided to used method 3 as it required neither preprocessing nor postprocessing and lead to the best translation results.

6.2. Case restoration

The 2006 evaluation conditions required the translation output to be in correct case ("truecasing"). As we lowercased the training corpus during the preprocessing in order to reduce the vocabulary size and improve the training, we needed to restore the correct case information. Therefore, we used the method described by the organizers of the evaluation and created a disambiguation language model. The model was based solely on the provided training data. Truecasing was done as a postprocessing step after the second pass of the translation using the disambiguation tool from the SRI language modeling toolkit [11]. Compared to the correct case of the DEV'06 references, truecasing had an error rate of 2%.

7. System combination

The system combination approach follows the description in [22]. For each input test sentence, the single-best translations of the partner systems are word-aligned with each other, allowing for word reordering. The alignment procedure is statistical and iterative. This procedure makes use of the fact that identical words should align to each other. The whole test corpus of translations is taken into account when determining the alignment.

When the mutual word alignment of all the hypotheses for one sentence is obtained, a primary hypothesis is selected. All other hypotheses are then reordered to match the word order of the primary hypothesis based on the alignment. Using the monotonic alignments of secondary hypotheses with the primary one, a confusion network is constructed. The consensus translation is then computed by "voting" on the confusion network, as in the ROVER approach of [23].

All arcs in the path through the confusion network representing a hypothesis of a particular MT system are weighted with a system-specific factor. The factors for the individual systems are optimized manually on the IWSLT 2006 Development set.

Since it is not known which hypothesis has the best word order, we let each hypothesis play the role of the primary translation once, and thus construct M confusion networks ($M = 4$ is the number of systems used) and unite them in a single lattice. Form the resulting lattice, the best hypothesis is extracted as the result of the system combination.

For the IWSLT 2006 evaluation, the system combination was performed on the output of the translation engines of the partners within the European TC-Star project: University of Karlsruhe, ITC-irst, RWTH Aachen University and Univer-

Table 5: Official results for the TC-Star submissions on the IWSLT 2006 Chinese-English test set.

Input	Accuracy Measures			Error Rates	
	BLEU [%]	NIST	Meteor [%]	WER [%]	PER [%]
Correct	24.1	6.40	51.8	65.4	49.8
Read	20.0	5.59	46.0	69.1	54.7

sitat Polytechnica de Catalunya (UPC). The submissions of the partners to the Chinese-to-English open data track were combined for text and read-speech input.

Table 5 shows the results of the TC-Star system combination submission. Compared to the best performing system within the combination (cf. 4), all measures except BLEU are improved by system combination, especially METEOR and PER. This can be explained by the fact, that the method used for system combination focuses primarily on improving the word choice rather than finding the correct reordering.

8. Conclusions

We have described the RWTH phrase-based statistical machine translation system that was used in the evaluation campaign of the IWSLT 2006. We use a two pass approach. In the first pass, we use a dynamic programming beam search algorithm to generate an N-best list. The second pass consists of rescoring and reranking of this N-best list.

One important advantage of our data-driven machine translation systems is that virtually the same system can be used for the different translation directions. Only a marginal portion of the overall performance can be attributed to language-specific methods.

We have shown significant improvements compared to the RWTH system of 2005 [1] and have introduced new feature functions based on phrase counts. New rescoring models were added in the second search path, the sentence mixture models and a sentence length model.

We also introduced a new method for punctuation prediction that uses the translation and language models to implicitly predict punctuation marks in the translation process.

9. Acknowledgement

This work was in part funded by the European Union under the integrated project TC-STAR – Technology and Corpora for Speech to Speech Translation (IST-2002-FP6-506738).

10. References

[1] R. Zens, O. Bender, S. Hasan, S. Khadivi, E. Matusov, J. Xu, Y. Zhang, and H. Ney, "The rwth phrase-based statistical machine translation system," in *Proceedings of the International Workshop on Spoken Language Translation (IWSLT)*, Pittsburgh, PA, October 2005, pp. 155–162.

[2] P. F. Brown, J. Cocke, S. A. Della Pietra, V. J. Della Pietra, F. Jelinek, J. D. Lafferty, R. L. Mercer, and P. S. Roossin, "A statistical approach to machine translation," *Computational Linguistics*, vol. 16, no. 2, pp. 79–85, June 1990.

[3] F. J. Och and H. Ney, "Discriminative training and maximum entropy models for statistical machine translation," in *Proc. of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, Philadelphia, PA, July 2002, pp. 295–302.

[4] F. J. Och, "Minimum error rate training in statistical machine translation," in *Proc. of the 41th Annual Meeting of the Association for Computational Linguistics (ACL)*, Sapporo, Japan, July 2003, pp. 160–167.

[5] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C++*. Cambridge, UK: Cambridge University Press, 2002.

[6] C. Tillmann and H. Ney, "Word reordering and a dynamic programming beam search algorithm for statistical machine translation," *Computational Linguistics*, vol. 29, no. 1, pp. 97–133, March 2003.

[7] R. Zens, H. Ney, T. Watanabe, and E. Sumita, "Reordering constraints for phrase-based statistical machine translation," in *COLING '04: The 20th Int. Conf. on Computational Linguistics*, Geneva, Switzerland, August 2004, pp. 205–211.

[8] R. Zens, F. J. Och, and H. Ney, "Phrase-based statistical machine translation," in *25th German Conf. on Artificial Intelligence (KI2002)*, ser. Lecture Notes in Artificial Intelligence (LNAI), M. Jarke, J. Koehler, and G. Lakemeyer, Eds., vol. 2479. Aachen, Germany: Springer Verlag, September 2002, pp. 18–32.

[9] F. J. Och, C. Tillmann, and H. Ney, "Improved alignment models for statistical machine translation," in *Proc. Joint SIGDAT Conf. on Empirical Methods in Natural Language Processing and Very Large Corpora*, University of Maryland, College Park, MD, June 1999, pp. 20–28.

[10] R. L. Graham, D. E. Knuth, and O. Patashnik, *Concrete Mathematics*, 2nd ed. Reading, Mass.: Addison-Wesley Publishing Company, 1994.

[11] A. Stolcke, "SRILM – an extensible language modeling toolkit," in *Proc. Int. Conf. on Spoken Language Processing*, vol. 2, Denver, CO, 2002, pp. 901–904.

[12] O. Bender, R. Zens, E. Matusov, and H. Ney, "Alignment Templates: the RWTH SMT System," in *Proc. of the Int. Workshop on Spoken Language Translation (IWSLT)*, Kyoto, Japan, September 2004, pp. 79–84.

[13] S. Hasan and H. Ney, "Clustered language models based on regular expressions for SMT," in *Proc. of the 10th Annual Conf. of the European Association for Machine Translation (EAMT)*, Budapest, Hungary, May 2005.

[14] R. M. Iyer and M. Ostendorf, “Modeling long distance dependence in language: Topic mixtures versus dynamic cache models,” *IEEE Transactions on Speech and Audio Processing*.

[15] F. J. Och, D. Gildea, S. Khudanpur, A. Sarkar, K. Yamada, A. Fraser, S. Kumar, L. Shen, D. Smith, K. Eng, V. Jain, Z. Jin, and D. Radev, “Syntax for statistical machine translation,” Johns Hopkins University 2003 Summer Workshop on Language Engineering, Center for Language and Speech Processing, Baltimore, MD, Tech. Rep., August 2003.

[16] R. Zens and H. Ney, “N-gram posterior probabilities for statistical machine translation,” in *Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL): Proceedings of the Workshop on Statistical Machine Translation*, New York City, NY, June 2006, pp. 72–77.

[17] 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 (LREC)*, Las Palmas, Spain, May 2002, pp. 147–152.

[18] F. J. Och and H. Ney, “A systematic comparison of various statistical alignment models,” *Computational Linguistics*, vol. 29, no. 1, pp. 19–51, March 2003.

[19] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, “Bleu: a method for automatic evaluation of machine translation,” in *Proc. of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, Philadelphia, PA, July 2002, pp. 311–318.

[20] G. Doddington, “Automatic evaluation of machine translation quality using n-gram co-occurrence statistics,” in *Proc. ARPA Workshop on Human Language Technology*, 2002.

[21] S. Banerjee and A. Lavie, “METEOR: An automatic metric for MT evaluation with improved correlation with human judgments,” in *43rd Annual Meeting of the Assoc. for Computational Linguistics: Proc. Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization*, Ann Arbor, MI, June 2005.

[22] E. Matusov, N. Ueffing, and H. Ney, “Computing consensus translation from multiple machine translation systems using enhanced hypotheses alignment,” in *Proceedings of EACL 2006 (11th Conference of the European Chapter of the Association for Computational Linguistics)*, Trento, Italy], month = April, year = 2006, pp. 33–40.

[23] J. Fiscus, “A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (rover),” in *IEEE Workshop on Automatic Speech Recognition and Understanding*, 1997.