The ITC-irst SMT System for IWSLT-2005

Boxing Chen, Roldano Cattoni, Nicola Bertoldi, Mauro Cettolo, Marcello Federico

ITC-irst - Centro per la Ricerca Scientifica e Tecnologica
I-38100 Povo - Trento, Italy

Abstract

This paper describes the statistical machine translation system developed at ITC-irst for the evaluation campaign of the International Workshop on Spoken Language Translation 2005. The system exploits two search passes: the first pass is performed by a beam-search decoder which generates an n-best list of translations, the second by a simple re-scoring algorithm. The two passes apply log-linear phrase-based models with an increasing number of feature functions. Runs have been submitted under the supplied-data and manual-transcription conditions for three language pairs: Chinese-to-English, Japanese-to-English and Arabic-to-English. Moreover, the Japanese-to-English system has been also employed under the ASR first-best condition. Significant improvements are reported by exploiting alternative word-alignments, and by using novel feature functions in the re-scoring step.

1. Introduction

This paper reports on the participation of ITC-irst in the evaluation campaign organized by the International Workshop on Spoken Language Translation (IWSLT) 2005.

One novelty with respect to the evaluation campaign of last year is that our Statistical Machine Translation (SMT) system, in addition to Chinese-to-English, has been applied to other two language-pairs: Japanese-to-English and Arabic-to-English. While for all language pairs manual transcripts were used as primary input, translations from Japanese to English were also computed by taking as input the ASR first-best output. For all the submitted runs, we have been working under the supplied-data condition, which constrains participants to develop their systems by using only data provided by the workshop organizers. While this condition boosts research on methods to cope with scarce language resources, it also permits to perform fair comparisons across systems from different sites.

This paper is organized as follows. Section 2 presents the general log-linear framework to SMT and overviews the architecture of our phrase-based SMT system. Section 3 provides details on our phrase extraction and model training methods. Section 4 outlines investigated techniques to improve system performance with limited training data. Finally, in Section 5 the experimental set-ups of the evaluation campaign runs and results are presented and discussed.

2. System Description

2.1. Log-Linear Model

Given a string $\mathbf{f}$ in the source language, the goal of SMT is to select the string $\mathbf{e}$ in the target language which maximizes the posterior probability $\Pr(\mathbf{e} \mid \mathbf{f})$. By introducing the hidden word *alignment* variable $\mathbf{a}$, the following approximate optimization criterion can be employed:

$$\begin{aligned} \mathbf{e}^* &= \arg\max_{\mathbf{e}} \Pr(\mathbf{e} \mid \mathbf{f}) \\ &= \arg\max_{\mathbf{e}} \sum_{\mathbf{a}} \Pr(\mathbf{e}, \mathbf{a} \mid \mathbf{f}) \\ &\approx \arg\max_{\mathbf{e},\mathbf{a}} \Pr(\mathbf{e}, \mathbf{a} \mid \mathbf{f}) \end{aligned} \quad (1)$$

By applying the maximum entropy [1] framework, the conditional distribution $\Pr(\mathbf{e}, \mathbf{a} \mid \mathbf{f})$ can be modeled through suitable real valued functions (called *feature functions*) $h_r(\mathbf{e}, \mathbf{f}, \mathbf{a}), r = 1 \ldots R$, and takes the parametric form:

$$\Pr(\mathbf{e}, \mathbf{a} \mid \mathbf{f}; \lambda) \propto \exp\{\sum_{r=1}^{R} \lambda_r h_r(\mathbf{e}, \mathbf{f}, \mathbf{a})\} \quad (2)$$

The ITC-irst system [2] is based on a log-linear model which extends the original IBM Model 4 [3] to *phrases*. In particular, target strings $\mathbf{e}$ are built from sequences of phrases $\tilde{e}_1 \ldots \tilde{e}_l$. For each target phrase $\tilde{e}$, the corresponding source phrase within the source string is identified through three random quantities: the *fertility* ϕ, which establishes its length; the *permutation* π_i, which sets the position of its first word; the *tablet* $\tilde{f}$, which defines its word string. Notice: target phrases might have fertility equal to zero, hence they do not translate any source word. Moreover, uncovered source positions are associated to a special target word (*null*), according to specific fertility and permutation random variables.
The resulting log-linear model applies eight feature functions, whose parameters are either estimated from data or empirically fixed:

- target trigram language model (estimated from monolingual texts)
- fertility model of target phrases (estimated from phrase-pair statistics, cf. Section 3)
- direct phrase-based lexicon model (as above)

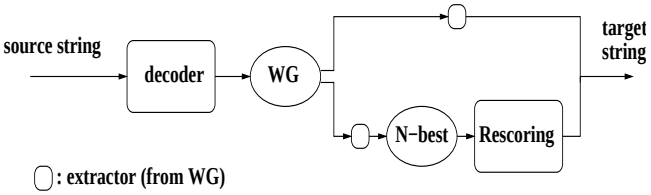

Figure 1: Decoding strategies: Given the word-graph (WG) produced by the decoder, either the 1-best translation is returned, or the N-best translations are extracted and re-scored with additional feature functions.

- inverse phrase-based lexicon model (as above)
- negative distortion model, for non-monotone coverage of source positions (negative exponential distribution)
- positive distortion model, for monotone coverage of source positions (as above)
- permutation model of `null` word (IBM Model 4)
- fertility model of `null` word (as above).

While feature functions exploit statistics extracted from the training data, the scaling factors λ_r of the log-linear model are estimated on the development data, by applying a *minimum error training* procedure [4], based on the simplex algorithm. A key role of this optimization process is played by the used metric. Initially, we tried to maximize the BLEU score, but noticed that the final output sentences were particularly short. Then, we tried to maximize with respect to the NIST score, which seems to exhibit an higher correlation with the human judgment of content adequacy. This choice increased sentence length, but at the cost of a significant deterioration of the BLEU score. A reasonable trade-off was finally obtained using the metric:

$$100 * BLEU + 4 * NIST$$

which appears to balance the contribution of both scores.

2.2. Decoding Strategy

Figure 1 illustrates how the translation of an input string is performed by the ITC-irst SMT system. In the first pass, a search algorithm (decoder) computes a word graph of translation hypotheses. Hence, either the best translation hypothesis is directly extracted from the word graph and output, or an N-best list of translations is computed [5]. The N-best translations are then re-ranked by applying additional feature functions (cf. Section 5.2) and the top ranking translation is finally output. Additional feature functions will be detailed in Section 5.2.

The decoder exploits a beam-search algorithm based on dynamic programming [11]. The optimal solution is computed by expanding and recombining previously computed partial *theories*. A theory is described by its *state*, which is the only information needed for its expansion. Expanded theories sharing the same state are recombined, i.e. only the one with the highest score is stored for further expansions. In order to output a word graph of translations, back-pointers to all expanded theories are maintained, too. Finally, N-best translations are extracted from the word-graph with an exact algorithm [5].

To cope with the large number of generated theories, approximations are introduced during the search:

Beam search: at each expansion less promising theories are pruned off by any of the following criteria:

- *threshold pruning*, i.e. the theory's score is lower than the current optimum score times a given threshold;
- *histogram pruning*: the theory's score is not among the top K best scores.

Both criteria are applied, with independent threshold settings, to all theories covering the same set of source positions, and to all theories with the same target string length.

Re-ordering constraints: at each theory-expansion step, a new source position is selected by limiting the number of vacant positions on the left-hand and the distance from the left-most vacant position. The maximum allowed values are called, respectively, maximum vacancy number (MVN) and maximum vacancy distance (MVD). Notice that consistent settings have MVN $\leq$ MVD and that the special case MVD=0 forbids word-reordering and makes the search monotone.

3. Phrase extraction and model training

Training of the phrase-based translation model exploits a parallel corpus provided with word-alignments in both directions, i.e. from source to target positions, and vice versa. This pre-processing step is accomplished by applying the GIZA++ software tool [9] (see details in section 5.1). Phrase-pair statistics are extracted as follows. Given a sentence pair $(\mathbf{f}, \mathbf{e})$, of lengths m and l, respectively, and its direct and inverted alignments $\mathbf{a}$ and $\mathbf{b}$, the union alignment is defined by:

$$\mathbf{c} = \{(j,i) : a_j = i \ \vee \ b_i = j\} \subseteq \{1,..m\} \times \{1,..,l\}.$$

Phrase-pairs are extracted from $(\mathbf{f}, \mathbf{e})$ which correspond to sub-intervals of the source and target positions, J and I, such that the union alignment $\mathbf{c}$ links all positions of J into I and all positions of I into J. In general, phrases are extracted with maximum length in the source and target defined by the parameters J_{max} and I_{max}. All such phrase-pairs are efficiently computed by an algorithm with complexity $\mathcal{O}(lI_{max}J^2_{max})$ [6].
More reliable phrase-pairs are obtained by filtering out pairs for which:

- lengths of source and target differ more than a factor 4;

- the following punctuation marks are not preserved between source and target phrases: period, open/closed parenthesis, question mark, quotation mark, and slash.

After the above filtering, the phrase-based lexicon and fertility models are estimated by applying the Witten-Bell smoothing method, as follows:

$$\Pr(\phi \mid \tilde{e}) = \frac{N(\phi, \tilde{e})}{N(\tilde{e}) + D(\tilde{e})} \quad (3)$$

$$\Pr(\tilde{f} \mid \phi, \tilde{e}) = \frac{N(\tilde{f}, \phi, \tilde{e})}{N(\phi, \tilde{e}) + D(\phi, \tilde{e})} \quad (4)$$

where $N(\cdot)$ indicates the number of occurrences, $D(\tilde{e})$ is the number of different ϕ observed with $\tilde{e}$, and $D(\phi, \tilde{e})$ is the number of different source phrases $\tilde{f}$ observed with ϕ and $\tilde{e}$. Inverted lexicon probabilities $\Pr(\tilde{e} \mid \tilde{f})$ are computed analogously.

For efficiency purposes, only the most probable target translations are considered for each source phrase, i.e. up to .95 of cumulative probability and up to 30 translations per phrase.

Target language models (LMs) used by the decoder and re-scoring modules are, respectively, estimated from 3-gram and 4-gram statistics by applying the *modified Kneser-Ney* smoothing method [7]. LMs are estimated with an in-house software toolkit which also provides a compact binary representation of the LM.

4. Model Training with Scarce Resources

The supplied-data condition represents a challenging task since training data are restricted to 20K sentences only. As statistical models tend to perform poorly with limited training data, different techniques were explored to overcome the data bottleneck.

Translation lexicon: a translation lexicon consists of pairs of source-target words which represent equivalent expressions with a high degree of reliability [8]. Translation pairs can be used to let a word-alignment tool produce higher-quality alignments. A translation lexicon was extracted by applying the Competitive Linking Algorithm (CLA) on the training data. The CLA [8] computes an association score between all possible word pairs within the parallel corpus, and then applies a greedy algorithm to select the best word-alignment for each sentence pair. The algorithm works under the one-to-one assumption, i.e. each source word can be aligned to one target word only, and vice versa. As a criterion for the extraction of translation equivalences, we used the frequency of word-pair alignments found by the CLA. Finally, the resulting word-pairs were added to the training data supplied to GIZA++.

Different word segmentations: in Asian languages there can be multiple ways of reasonably segmenting a sequence of characters into words. IWSLT-05 supplied data are all segmented, probably by humans. We re-segmented the training data by an algorithm which only exploits the word frequencies of the original corpus. The re-segmented corpus was added to the original training corpus as an additional resource. We used this approach for the Chinese-to-English and Japanese-to-English translation tasks.

Sentence splitting: long parallel sentences can be split into shorter and aligned portions (chunks) by exploiting IBM Model 1 statistics computed on the original corpus [6]. The chunked sentences can be then added to the original training corpus as an additional resource.

Additional word alignments: extraction of phrases relies on word alignments (see Section 3). In addition to alignments provided by the GIZA++ toolkit, we exploited word-alignments provided by the CLA. In particular, phrase-pairs extracted from the CLA alignments were added to those extracted in the usual way. As a result, the original set of phrase-pair was significantly extended.

Table 1: Statistics for the Chinese-to-English Supplied Data track after pre-processing.

		Chinese	English
train	sentences	20,000	
	running words	173,103	181,641
	vocabulary	8,536	7,405
	singletons	3,959	3,249
dev1	sentences	506	8,096
(CSTAR-03)	running words	3,514	65,615
dev2	sentences	500	8,000
(IWSLT-04)	running words	3,806	64,884
test	sentences	506	
	running words	3,795	–

Table 2: Statistics for the Japanese-to-English Supplied Data track after pre-processing.

		Japanese	English
train	sentences	20,000	
	running words	171,259	181,931
	vocabulary	9,251	7,348
	singletons	4,411	3,216
dev1	sentences	506	8,096
(CSTAR-03)	running words	3,531	65,615
dev2	sentences	500	8,000
(IWSLT-04)	running words	3,538	64,884
test	sentences	506	
	running words	4,226	–

Table 3: Statistics for the Arabic-to-English Supplied Data track after pre-processing.

		Arabic	English
train	sentences	20,000	
	running words	159,307	182,234
	vocabulary	18,150	7,344
	singletons	10,092	3,215
dev1	sentences	506	8,096
(CSTAR-03)	running words	3,259	65,615
dev2	sentences	500	8,000
(IWSLT-04)	running words	3,359	64,884
test	sentences	506	
	running words	3,252	–

5. Experiments

Experiments were carried out in the context of the *Basic Traveling Expression Corpus* (BTEC) task [10]. BTEC is a multilingual speech corpus which contains translation pairs coming from phrase books for tourists. Participants in the IWSLT 2005 evaluation were provided with the following data for each language-pair:

- a training set of 20,000 parallel sentences;
- two development sets of 506 (C-STAR 2003) and 500 (IWSLT 2004) sentences, respectively, provided with multiple reference translations;
- the IWSLT 2005 test set of 506 sentences (source language only).

Detailed figures about the employed data sets are reported in Tables 1-3. ITC-irst took part in the supplied-data track only. Hence, system training was only based on the provided 20,000 sentence pairs. System tuning exploited the CSTAR 2003 development set. It is worth underlying that the development set was only used to estimate the *weights* of the features functions, while the features functions themselves were estimated on the training set only. Finally, the IWSLT 2004 development set was employed as a blind test to check system performance.

Concerning the other evaluation conditions, we evaluated our system on Chinese-to-English, Japanese-to-English and Arabic-to-English with human transcriptions as input. Additionally, the Japanese-to-English system was also evaluated by feeding it with the ASR first-best output.

5.1. Preprocessing

Preprocessing aims at normalizing source and target texts in order to reduce data sparseness. Preprocessing differs according to the specific language-pair taken into account.

Chinese-to-English: the source sentences are first converted from GB into the CKZ format which is a full ASCII encoding. Then, word spaces are deleted and sentences are re-segmented into words. Re-segmentation is performed by an in-house dynamic-programming tool that relies on word occurrence statistics collected from the supplied training corpus. In this way, inconsistencies in human segmentation are possibly smoothed away and the segmentation results more uniform – experiments on baselines confirmed this outcome. After a tokenization step that separates words from punctuation, numbers written in textual form are transformed into digits. The same preprocessing steps are applied to the target sentences except for word segmentation; in addition, words are put in lower case. Finally, parallel sentences are discarded if source and target differ too much in length.

Japanese-to-English: as for Chinese, the source sentences are first converted from GB into the CKZ format. No additional word segmentation is performed on Japanese, since experiments on baselines suggest that the original segmentation works better than the new one. Moreover, no processing is applied to manage punctuation marks and numbers. The standard pre-processing is finally applied to the English portion of the corpus followed by the length-based filtering described above.

Arabic-to-English: the source sentences are first converted from UTF-8 into a simple full ASCII encoding – experiments on baselines show a little gain in performance when using such encoding. Punctuation is separated from words by a tokenization tool kindly provided by RWTH, Aachen, Germany. For the Arabic language no other pre-processing is applied. Accordingly, the English sentences are only tokenized and put in lower case. Finally the filtering step on the parallel sentences is applied.

Word-alignments from source to target words, and vice-versa, were computed on the preprocessed training corpora by means of two different techniques: the GIZA++ toolkit [9], which provides Viterbi alignments based on IBM Model 4, and the CLA (cf. Section 4).

5.2. Optimization

To set-up the baselines, the maximum length of the phrases (I_{max} and J_{max}) was set to 8 and monotone search was applied.

Optimization was carried out on the CSTAR-03 development set by applying different techniques in an incremental way, i.e.: translation lexicon, addition of CLA word-alignments, re-segmentation at word level, data chunking, and non-monotone search. Finally, a re-scoring step with nine additional feature functions was applied to the 1000-best translations.

5.2.1. Non-monotone Search

Japanese and English differ sensibly in the word order. Unfortunately, phrases may capture long-distance word re-

Table 4: Results of the optimization techniques on the CSTAR-03 development set (BLEU% score).

System	Chi2Eng	Jap2Eng	Ara2Eng
baseline	31.79	35.09	54.23
+translation lexicon	32.60	38.15	56.41
+additional alignments	34.20	39.83	57.73
+re-segmented data	34.33	39.85	–
+chunked data	–	40.14	–
+non-monotone search	38.71	45.84	60.16
+re-scoring	45.22	52.75	61.81

Table 5: Results of the optimization techniques on the IWSLT-04 development set (BLEU% score).

System	Chi2Eng	Jap2Eng	Ara2Eng
baseline	35.82	33.82	51.01
+translation lexicon	36.28	35.78	52.84
+additional alignments	37.59	38.77	54.14
+re-segmented data	38.29	38.97	–
+chunked data	–	39.59	–
+non-monotone search	42.51	44.66	56.40
+re-scoring	47.99	51.01	57.94

ordering only to some extent. Hence, re-ordering constraints set in the search algorithm can be relaxed, at the cost of decoding time and memory. Experiments were conducted in order to find a reasonable trade-off between translation accuracy and time.

Table 6 reports BLEU score results on Japanese-to-English, by varying the word-reordering parameters MVD and MVN introduced in Section 2.2. In particular, the MVD versus the difference MVD-MVN was considered. Best performance were obtained with MVD set to 7, and the difference MVD-MVN between 0 and 1. According to these experiments, we adopted the setting MVD=7 and MVN=6 for Japanese-English.

Additional experiments on non-monotone search decoding with the other two language pairs suggested to adopt for them the setting MVD=6 and MVN=5.

5.2.2. Re-scoring

The following nine additional feature functions were applied to re-score each of the 1000-best translation hypotheses:

- IBM model 1 lexicon score, over all possible alignments
- IBM model 3 lexicon score, over all possible alignments

Table 6: BLEU% scores on the Japanese-to-English baseline with different settings of the word-reordering parameters of the search algorithm. MVD and MVN stand for maximum vacancy distance and maximum vacancy number, respectively.

	MVD				
MVD-MVN	4	5	6	7	8
0	43.65	43.60	43.97	44.30	
1	43.54	43.61	43.75	44.27	43.89
2		43.84	43.71	43.85	43.73
3			44.06	43.94	44.04
4				43.47	43.78
5					43.48

- CLA lexicon score, over all possible alignments
- question feature, i.e. a binary feature which triggers when the text ends with a question mark and starts with one of the typical starting words of question sentences found in training data
- frequency of its n-grams (n=1,2,3,4) within the 1000-best translations
- ratio of the target length and source length
- 2-grams target language model
- 4-grams target language model
- 5-grams target language model.

5.2.3. Performance Discussion

The results of the single optimization steps are shown in Table 4 and Table 5. Notice that all translation scores reported in this work are computed on texts with punctuation and no case information. A significant contribution in performance was given by the use of the CLA. By putting together the effect of the translation lexicon and the addition of CLA word-alignments to the training data, BLEU score improved on the IWSLT 2004 dev set between 5% and 15% relative. Remarkably, for the Japanese-to-English task, the BLEU% score rised from 33.82% to 38.77%. Given that these languages have very different word order, it seems that use of alternative word-alignment models can be beneficial to the quality of phrase-pair statistics.

The use of additional word-segmentations to smooth out inconsistencies in the manual segmentations showed to be more effective for Chinese than for Japanese data. Parallel text chunking was only applied to Japanese-English as an attempt to help the work of the word-alignment models, given the difficulty of word-reordering between the two languages.

The application of non-monotone search gave major improvements in performance: 11% relative BLEU improvement for Chinese (from 38.28 to 42.51), 13% for Japanese

Table 7: Contribution of each feature function in the re-scoring step on the CSTAR-03 development set (BLEU% score).

System	Chi2Eng	Jap2Eng	Ara2Eng
baseline	38.71	45.84	60.16
IBM model-1	38.81	45.75	59.45
IBM model-3	37.44	46.11	59.55
CLA score	38.20	45.98	60.66
question tag	39.45	46.89	60.42
n-grams	40.79	47.39	60.41
target length	38.25	42.22	55.03
2-grams LM	38.93	45.61	60.55
4-grams LM	41.98	49.29	60.80
5-grams LM	41.26	48.83	61.17
+all features	45.22	52.72	61.81

Table 8: Contribution of each feature function in the re-scoring step on the IWSLT-04 development set (BLEU% score).

System	Chi2Eng	Jap2Eng	Ara2Eng
baseline	42.51	44.66	56.40
IBM model-1	42.31	44.48	56.00
IBM model-3	41.53	44.97	56.16
CLA score	42.42	45.20	56.31
question tag	42.81	45.83	56.66
n-grams	43.71	46.19	56.89
target length	41.11	41.00	50.87
2-grams LM	44.06	45.34	56.07
4-grams LM	45.88	45.51	56.72
5-grams LM	45.72	45.81	56.61
+all features	47.99	51.01	57.94

(from 39.59 to 44.66), and 4% for Arabic (from 54.14 to 56.40). As expected, non-monotone search is more effective for language pairs with very different grammatical structures, which makes indeed SMT more difficult.

Re-scoring provides an additional boost in performance. Tables 7 and 8 show the single contribution of the 9 feature functions used in re-scoring. Besides the feature functions which are well known in the literature, two novel feature functions provided a significant improvement in the BLEU score. The "question tag" feature function increased the BLEU score by 2% relative for Chinese and Japanese, while the "n-gram" feature function increased the BLEU score by 5% and 3% on Chinese and Japanese, respectively. Lower gains can be observed for Arabic-English, for which it seems more difficult to improve the already high performance of the baseline system.

It is worth noticing that despite some feature functions decrease the baseline score, every feature function ensures a positive contribution to the final performance. In other words, removal of any feature function resulted in a decrease of the BLEU score.

5.3. Official Results

5.3.1. Development sets

Tables 9 and 10 show the official scores on the development sets CSTAR-03 and IWSLT-04 as computed by the IWSLT-05 submission server. The BLEU scores slightly differ from those computed with our in-house tool (cf. Table 7 and Table 8). This is mainly because the official IWSLT software applies some pre-processing on the translations before computing the scores, e.g. it removes all punctuation marks. Nevertheless, relative differences among the submitted systems are confirmed. Also the other metrics computed by the evaluation server maintain the original ranking.

Table 9: Official scores on the CSTAR-03 development set (supplied data and manual transcription).

	BLEU%	NIST	WER	PER	METEOR
Chi2Eng	44.84	7.51	47.02	43.75	61.47
Jap2Eng	53.19	8.75	43.75	35.96	68.15
Ara2Eng	61.19	9.89	33.64	29.80	74.20

5.3.2. Test set

Table 11 shows the official results on the test set as reported by the submission server. The following remarks can be made. On the manual transcriptions, scores of the Japanese-to-English system are significantly lower than expected. On both dev sets this language pair performed much better than Chinese-to-English. An explanation is suggested by the high score of the Japanese-to-English ASR system. Past experiments indicated that the degradation due to the ASR errors corresponds to a loss of around 10% in the BLEU score. A closer look at the test set of the manual transcription revealed a particularly high out-of-vocabulary (OOV) rate: more than 20%. It is definitely much higher than the OOV rates of the development sets (less than 2%). Indeed, the OOV rate of

Table 10: Official scores on the IWSLT-04 development set (supplied data and manual transcription).

	BLEU%	NIST	WER	PER	METEOR
Chi2Eng	46.37	8.32	47.90	40.17	63.90
Jap2Eng	50.11	8.99	46.77	36.88	66.93
Ara2Eng	56.37	9.58	35.97	31.35	73.03

Table 11: Official scores on the IWSLT-05 test set (Supplied Data tracks).

	Input	BLEU%	NIST	WER	PER	METEOR	GTM
Chi2Eng	manual	52.75	9.0598	41.36	34.56	68.93	62.00
Jap2Eng	manual	43.13	7.0983	51.58	43.52	58.67	49.16
	ASR 1-best	42.95	8.2684	50.73	41.90	61.83	50.43
Ara2Eng	manual	56.22	9.6572	36.83	31.28	73.16	66.85

ASR first-best transcriptions is more reasonable, i.e. around 1.5%, which explains why performance is better than on manual transcriptions.

Finally, we underline the progress made during the last year. On the 2004 Chinese-English test set, the BLEU% performance of our system increased from 35.88 to 47.99. Moreover, official results of IWSLT 2005, on all considered language pairs, show that the ITC-irst system is among the top performing ones.

6. Conclusions

This paper described the ITC-irst SMT system developed for the IWSLT 2005 evaluation campaign. The system is based on a two-pass beam-search decoder exploiting a phrase-based log-linear model. In the first decoding step, a limited number of local feature functions is employed and n-best translation hypotheses are generated. In the second step, a larger set of local and global feature functions is employed to re-rank the n-best translations. Significant improvements with respect to the 2004 IWSLT systems were obtained mainly from the use of a larger number of feature functions and an additional word-alignment method, namely the competitive linking algorithm.

7. Acknowledgments

This work has been funded by the European Union under the integrated project TC-STAR - Technology and Corpora for Speech to Speech Translation - (IST-2002-FP6-506738, http://www.tc-star.org).

8. References

[1] A. Berger, S. Della Pietra, and V. Della Pietra, "A Maximum Entropy Approach to Natural Language Processing", *Computational Linguistics*, vol. 22, no. 1, pp. 39–71, 1996.

[2] N. Bertoldi, R. Cattoni, M. Cettolo, and M. Federico, "The ITC-irst Statistical Machine Translation System for IWSLT-2004", in *Proceedings of IWSLT*, Kyoto, Japan, 2004.

[3] P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, and R. L. Mercer, "The Mathematics of Statistical Machine Translation: Parameter Estimation", *Computational Linguistics*, vol. 19, no. 2, pp. 263–313, 1993.

[4] M. Cettolo and M. Federico, "Minimum Error Training of Log-Linear Translation Models", in *Proceedings of IWSLT*, Kyoto, Japan, 2004.

[5] B. H. Tran, F. Seide, and V. Steinbiss, "A Word Graph based N-Best Search in Continuous Speech Recognition", in *Proceedings of ICSLP*, Philadelphia, PA, USA, 1996.

[6] M. Cettolo, M. Federico, N. Bertoldi, R. Cattoni, and B. Chen, "A Look inside the ITC-irst SMT System", in *Proceedings of the 10th MT-Summit*, Phuket, Thailand, 2005.

[7] J. Goodman and S. Chen, "An empirical study of smoothing techniques for language modeling", Harvard University, Technical Report TR-10-98, August 1998.

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

[9] F. J. Och and H. Ney, "Improved Statistical Alignment Models", in *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, Hong Kong, China, 2000.

[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 *Proceedings of 3rd International Conference on Language Resources and Evaluation (LREC)*, Las Palmas, Spain, 2002, pp. 147–152.

[11] 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, 2003.