The TALP Ngram-based SMT System for IWSLT 2007

Patrik Lambert, Marta R. Costa-jussà, Josep M. Crego, Maxim Khalilov, José B. Mariño, Rafael E. Banchs, José A.R. Fonollosa and Holger Schwenk[1]

TALP Research Center
Universitat Politècnica de Catalunya, Barcelona
{lambert|mruiz|jmcrego|khalilov|canton|rbanchs|adrian}@talp.upc.edu

[1] LIMSI-CNRS, BP 133
91403 Orsay Cedex
schwenk@limsi.fr

Abstract

This paper describes **TALPtuples**, the 2007 N-gram-based statistical machine translation system developed at the TALP Research Center of the UPC (Universitat Politècnica de Catalunya) in Barcelona. Emphasis is put on improvements and extensions of the system of previous years. Mainly, these include optimizing alignment parameters in function of translation metric scores and rescoring with a neural network language model.

Results on two translation directions are reported, namely from Arabic and Chinese into English, thoroughly explaining all language-related preprocessing and translation schemes.

1. Introduction

Ngram-based Machine Translation (MT), originally based in the Finite-State Transducers approach to Statistical MT (SMT) [1, 2], has proved to be a competitive alternative to phrase-based and other state-of-the-art systems in previous evaluation campaigns, as shown in [3, 4].

Efforts have been focused on improving translation according to human evaluation by further developing different stages of the SMT system: alignment and rescoring.

As in previous years, we aligned the training corpus using Giza++ software. However, instead of keeping the default parameters, we performed a minimum translation error training procedure to adjust Giza++ smoothing parameters to the task. This procedure had been successful with an alignment system based on discriminative training [5].

For the rescoring we incorporate a neural network language model as previously experienced in [6]. The neural network language model mainly is able to produce a better generalization in the translation system.

This paper is organized as follows. Section 2 briefly reviews last year's system, including tuple definition and extraction, translation model and feature functions, decoding tool and reordering and optimization criterion. Section 3 describes the alignment translation-minimum-error training procedure. Section 4 focuses on rescoring using a neural language model (NNLM). Next, Section 5 reports on all experiments carried out from Arabic and Chinese into English for IWSLT 2007. Finally, Section 6 sums up the main conclusions from the paper.

2. Baseline description

2.1. N-gram-based Machine Translation

The TALP Ngram-based SMT system performs a log-linear combination of a translation model and additional feature functions (see further details in [7, 8]). In contrast to phrase-based models, our translation model is estimated as a standard n-gram model of a bilingual language expressed in *tuples*. In this way, it approximates the joint probability between source and target languages capturing bilingual context, as described by the following equation:

$$p(S,T) = \prod_{k=1}^{K} p((\tilde{s},\tilde{t})_k | (\tilde{s},\tilde{t})_{k-N+1}, ..., (\tilde{s},\tilde{t})_{k-1}) \quad (1)$$

where s refers to source, t to target, and $(\tilde{s},\tilde{t})_k$ to the k^{th} tuple of a given bilingual sentence pair segmented in K tuples.

2.2. Tuple extraction

Given a certain word-aligned parallel corpus, tuples are extracted according to the following constraints [9]:

- a monotonic segmentation of each bilingual sentence pair is produced
- no word in a tuple is aligned to words outside of it
- no smaller tuples can be extracted without violating the previous constraints

However, when dealing with pairs of languages with non-monotonic word order, a certain reordering strategy is required to extract more reusable units (less sparse). Hence, we allow the source words to be reordered before extracting translation units from training sentence pairs by following the word-to-word alignments. The unfolding technique is fully described in [10].

Figure 1 shows an example of tuple unfolding compared to the monotonic extraction. The unfolding technique produces a different bilingual n-gram language model with reordered source words.

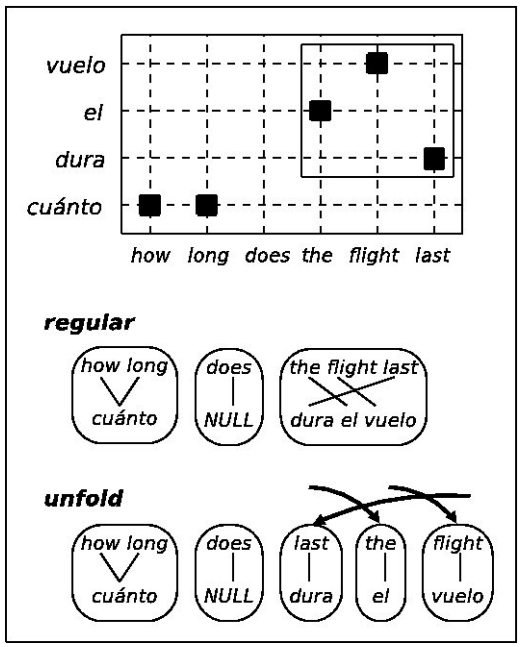

Figure 1: *Comparing regular and unfolded tuples.*

The unfold method needs the input source words be reordered during decoding similarly to how source words were reordered in training. If monotonic decoding were used with unfolded units, translation hypotheses would be formed following the source language word order. The reordering approach used in this work is detailed in section 2.6.

2.3. Feature functions

As additional feature functions to better guide the translation process, the system incorporates six models: a target language model, a word bonus model, two lexicon models, a target (part-of-speech) tagged language model and a source (part-of-speech) tagged language model.

The *target language model* (target LM) is estimated as a standard n-gram over the target words, as follows:

$$p_{LM}(T) \approx \prod_{n=1}^{N} p(t_n|t_{n-2}, t_{n-1}) \qquad (2)$$

where t_n refers to the n^{th} word in the partial translation hypothesis T.

Usually, this feature is accompanied by a *word bonus model* based on sentence length, compensating the target language model preference for short sentences (in number of target words). This bonus depends on the number of target words in the partial hypothesis, denoted as:

$$p_{WP}(T) = exp(\text{number of words in } T). \qquad (3)$$

The third and fourth feature functions correspond to source-to-target and target-to-source *lexicon models*. These models use IBM model 1 translation probabilities to compute a lexical weight for each tuple, accounting for the statistical consistency of the pairs of words inside the tuple. These lexicon models are computed according to the following equation:

$$p_{IBM1}((\tilde{s}, \tilde{t})_k) = \frac{1}{(I+1)^J} \prod_{j=1}^{J} \sum_{i=0}^{I} p(t_k^i|s_k^j) \qquad (4)$$

where s_k^j and t_k^i are the j^{th} and i^{th} words in the source and target sides of tuple $(\tilde{s}, \tilde{t})_k$, being J and I the corresponding total number of words in each side of it.

To compute the forward lexicon model, IBM model 1 lexical parameters from GIZA++ source-to-target alignments are used. In the case of the backward lexicon model, GIZA++ target-to-source alignments are used instead.

The *target tagged language model* is estimated as a standard N-gram LM. It aims at achieving generalization power over the target side words.

Finally, the *source tagged language model* is also estimated as a standard N-gram LM. It is computed over the source side POS tags after being reordered. Hence, aiming at describing the reordering process introduced in training.

2.4. MARIE decoder

As decoder, we use MARIE [11], a beam-search decoder which taking the previous models into account developed at TALP Research Center. For efficient pruning of the search space, *threshold pruning*, *histogram pruning* and *hypothesis recombination* are used.

Apart from monotone search, MARIE also implements full reordered search, which can be constrained by a set of parameters, as explained in the following section.

The primary TALPtuples systems did not incorporate any rescoring module, therefore choosing their 1-best hypothesis as final translation solution. Nevertheless, for the Chinese-English task, a secondary run was performed with a rescoring module, as described in Sections 4 and 5.3.2.

2.5. Feature Weights Optimization

To tune the weight of each feature function in the SMT system, we used the Simultaneous Perturbation Stochastic Approximation (SPSA) algorithm [12]. SPSA is a stochastic

implementation of the conjugate gradient method which requires only two evaluations of the objective function in each iteration, regardless of the dimension of the optimization problem. It was observed to be more robust than the Downhill Simplex method when tuning SMT coefficients [13]. The SPSA procedure is in the general recursive stochastic approximation form:

$$\hat{\lambda}_{k+1} = \hat{\lambda}_k - \mathbf{a}_k \hat{\mathbf{g}}_k(\hat{\lambda}_k) \quad (5)$$

where k here refers to the iteration number, $\hat{\mathbf{g}}_k(\hat{\lambda}_k)$ is the estimate of the gradient $\mathbf{g}(\lambda) \equiv \partial E/\partial\lambda$ at the iterate $\hat{\lambda}_k$ based on the previous mentioned evaluations of the objective function. a_k denotes a positive number that usually gets smaller as k gets larger.

Two-sided gradient approximations involve evaluations of $E(\hat{\lambda}_k + \text{perturbation})$ and $E(\hat{\lambda}_k - \text{perturbation})$.

In the simultaneous perturbation approximation, all elements of $\hat{\lambda}_k$ are randomly perturbed together and the approximated gradient vector is:

$$\frac{E(\hat{\lambda}_k + c_k\mathbf{\Delta_k}) - E(\hat{\lambda}_k - c_k\mathbf{\Delta_k})}{2c_k} \begin{bmatrix} 1/\Delta_{k1} \\ 1/\Delta_{k2} \\ \vdots \\ 1/\Delta_{kN} \end{bmatrix} \quad (6)$$

In equation 6, $\mathbf{\Delta}_k$ is a perturbation vector of same dimension N as λ, whose values Δ_i are computed randomly. c_k denotes a small positive number that usually gets smaller as k gets larger. Notice that in general, SPSA converges to a local minimum.

Two optimization schemes are possible. In the first one, the development corpus is translated at each iteration. With 6 parameters (one parameter can remain fixed to 1, the others being scaled accordingly), the algorithm converges after about 60 to 100 iterations. Thus, in this scheme, in the order of 80 development corpus translations are required. In the second scheme, an N-best list is produced by the decoder. The optimization algorithm is used to minimize the translation error while rescoring this N-best list. With the optimal coefficients, a new decoding is performed so as to produced an updated N-best list [14]. This process converges after only 5 to 10 decodings. For each internal optimization, about 80 iterations are still required, but each iteration is much shorter since it only requires to rescore an N-best list.

We used the second scheme with $\frac{1}{2}$(BLEU+NIST) as maximization criterion.

2.6. Reordering Strategies

The reordering framework followed in this work consists of using a set of automatically learned rewrite rules to extend the monotonic search graph with reordering hypotheses (details in [15]).

Patterns are extracted in training from the crossed links found in the word alignment, in other words, found in translation tuples (as no word within a tuple can be linked to a word out of it [9]).

Starting from the monotonic graph, each sequence of input POS tags fulfilling a source-side rewrite rule implies the addition of a reordering arc (which encodes the reordering detailed in the target-side of the rule). Figure 2 shows how three rewrite rules applied over an input sentence extend the search graph given the reordering patterns that match the source POS tag sequence [1].

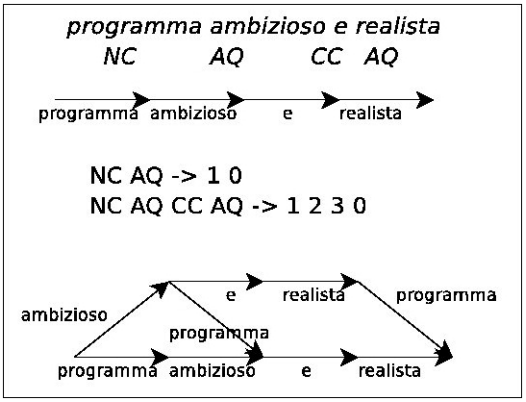

Figure 2: *Search graph extension*.

In the search, the decoder makes use of the whole set of models to score each reordering hypothesis, mainly driven by the N-gram translation model, as it has been estimated with reordered source words.

3. Alignment Minimum Translation Error Training

Alignment smoothing parameters were tuned via the optimization procedure depicted in Figure 3.

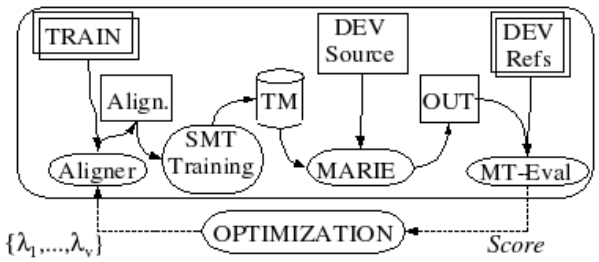

Figure 3: *Optimization loop*.

The training corpus was aligned with a set of initial parameters $\lambda_1, \ldots, \lambda_\nu$. This alignment was used to extract tuples and build a bilingual N-gram translation model (TM). A basic SMT system, consisting of MARIE decoder and this translation model as single feature[2], was used to produce

[1] *NC*, *CC* and *AQ* stand respectively for name, conjunction and adjective.

[2] An N-gram SMT system can produce good translations without additional target language model since the target language is modeled inside the bilingual N-gram model.

a translation (OUT) of the development source set. Then, translation quality over the development set is maximized by iteratively varying the set of coefficients.

The optimization procedure was performed by using the SPSA algorithm, described in Section 2.5. Each function evaluation required to align the training corpus and build a new translation model. The algorithm converged after about 50-80 evaluations.

Finally, the corpus was aligned with the optimum set of coefficients. Translation units were extracted from this alignment.

4. Neural Network Language Model

The basic idea of the continuous space LM, also called neural network LM, is to project the word indexes onto a continuous space and to use a probability estimator operating on this space. Since the resulting probability functions are smooth functions of the word representation, better generalization to unknown n-grams can be expected. This is believed to be particularly important for tasks with limited resources, as it is the case for IWSLT. A neural network can be used to simultaneously learn the projection of the words onto the continuous space and to estimate the n-gram probabilities. This is still a n-gram approach, but the LM posterior probabilities are "interpolated" for any possible context of length n-1 instead of backing-off to shorter contexts. For more details on this approach, see [6] and references there in.

5. Experiments

In this section the experimental work conducted for IWSLT 2007 shared tasks is reported. UPC participated in the Arabic to English and the Chinese to English tasks.

5.1. Tasks Description

Although this year all publicly available data was allowed, we only used the provided data to train our system. Our internal training data consisted in the provided training data plus dev1, dev2 and dev3 sets. Only dev sets sentence pairs containing the first English reference were added to the bilingual training data, whereas all English references were added to the monolingual data[3]. True case and punctuation marks were removed from these training data. Punctuation marks and true case were restored by using SRILM 'disambig' tool as suggested by IWSLT organizers. System coefficients were tuned with dev4 set and dev5 was used as an internal test set. Both dev4 and dev5 contained punctuation marks and true case. After obtaining the final configuration, dev4 and dev5 were added to the training data in the same way as dev1, dev2 and dev3 were previously added, and the final system was trained.

In both Arabic to English and Chinese to English tasks, the 1-best speech recognition output was taken as input to the translation system. Therefore, no n-best list nor word graph were used as input.

Tables 1 and 2 show corpora statistics for both language pairs. Number of sentences, running words, vocabulary, sentence length and human references are indicated. Bilingual and monolingual corpora statistics are shown for development and final training data.

		sent.	wrds	voc.	slen.
devel bil train	ar	24.4k	189k	10.9k	7.7
	en		170k	6.9k	7.0
devel monol train	en	71.0k	492k	9.7k	6.9
final bil train	ar	25.4k	201k	11.3k	7.9
	en		182k	7.1k	7.2
final monol train	en	77.9k	578k	10.1k	7.4
dev4	ar	489	5912	1224	12.1
dev5	ar	500	6579	1481	13.2

Table 1: *Arabic→English corpus statistics.*

		sent.	wrds	voc.	slen.
devel bil train	zh	47.3k	318k	9.8k	6.7
	en		331k	9.0k	7.0
devel monol train	en	71.0k	492k	9.7k	6.9
final bil train	zh	48.3k	329k	9.9k	6.8
	en		343k	9.2k	7.1
final monol train	en	77.9k	578k	10.1k	7.4
dev4	zh	489	5476	1094	11.2
dev5	zh	500	5846	1292	11.7

Table 2: *Chinese→English corpus statistics.*

5.2. Data Preprocessing

For all language pairs, training sentences were split by using final dots on both sides of the bilingual text (when the number of dots was equal), increasing the number of sentences and reducing its length. Specific preprocessing for each language is detailed in the following respective section.

5.2.1. Arabic

Following a similar approach to that in [16], we used the MADA+TOKAN system for disambiguation and tokenization. For disambiguation only diacritic uni-gram statistics were employed. For tokenization we used the D3 scheme with -TAGBIES option. The D3 scheme splits the following set of clitics: w+, f+, b+, k+, l+, Al+ and pronominal clitics. The -TAGBIES option produces Bies POS tags on all taggable tokens.

[3]When all references are added to both bilingual and monolingual data, BLEU score is improved but METEOR score gets worse. Since in this task BLEU score is well correlated to fluency and METEOR is well correlated to adequacy [4], we supposed that adding all references was beneficial to monolingual language models but not to the bilingual language model.

5.2.2. Chinese

Chinese preprocessing included re-segmentation using ICTCLAS [17] and POS tagging using the freely available Stanford Parser[4].

5.2.3. English

English preprocessing includes Part-Of-Speech tagging using freely-available *TnT* tagger [18].

For alignment purpose only (of the ZhEn system), the English corpus was stemmed using the Snowball stemmer [5], based on Porter's algorithm.

5.3. Results

5.3.1. Alignment

In the ZhEn system development work, we tried to improve word alignment by stemming the English corpus and make use of classes [19]. We also performed several combinations of source-target and target-source GIZA++ alignments (union, growing forward diagonal method and Och's refined method [20]), as well as concatenations of various of these combinations. Using stems and classes in the alignment improved translation results in all cases, and the best combination for the system with pattern-based reordering was the union[6]. At the end, the best alignment configuration for our baseline system was obtained with Giza++ software, running respectively 5, 5, 3 and 3 iterations of models 1, HMM, 3 and 4, using English stems and 50 classes and taking the union of source-target and target-source alignments.

Table 3 show results for the new features of this year's system.

We optimized the following GIZA++ parameters by means of the minimum translation error training (MET) procedure described in section 3: smoothing factors for models HMM, IBM3 and IBM4, as well as the probability for the empty word. Notice that the empty word plays an important role in our translation model, so tuning this parameter may have some impact. We performed an optimization of these parameters in function of machine translation score for each value of the deficient distortion for empty word (defdisEmpty) parameter (0, 1 and 2). Then we aligned the corpus with the optimal parameters, built the SMT system, and evaluated it. Among the three optimizations, only the one performed with $defdisEmpty = 1$ yielded an improvement in both dev and test sets. The corresponding results are shown in table 3.

5.3.2. Rescoring

In this work, the continuous space LM was trained on the same data than the back-off LM. The design parameters for the neural networks are as follows. The hidden layer was of dimension 200 and the output layer was limited to the 8192 most frequent words (short list). As in previous works, several neural networks with different sizes of the projection layer were trained and interpolated, together with the back-off LM. The interpolation coefficients were optimized on the development data using an EM procedure.

Incorporation into the SMT system was done using 1000-best lists. After replacing the LM scores in the n-best list the feature-function coefficients were tuned again.

Table 3 summarizes the results obtained when the continuous space LM (NNLM) was used. After optimization of the overall system, a continuous space LM was trained on all the available data, including Dev4 and Dev5, using the same settings of the various parameters and coefficients.

Notice that Table 3 shows only results for the Chinese-English task, because the new features of this year's system have only been applied to that system.

5.3.3. Official Evaluation Results

In this section we report the BLEU scores obtained in the official evaluation for Arabic to English and Chinese to English tasks.

	UPC	Best	Rank
AE ASR Primary	0.4445	0.4445	1/11
AE Clean Primary	0.4804	0.4923	3/11
CE Clean Primary	0.2991	0.4077	11/15
CE Clean Primary + NNLM	0.2920	0.4077	-

Table 4: *Official translation results (BLEU scores) for IWSLT 2007 Chinese-English and Arabic-English tasks. Next to our system's score, we indicated the Best system's score. For the primary runs, we also indicated the rank of our system among all primary runs.*

The Arabic translation scores show that our system is able to achieve excellent results in this type of task, compared to other systems. It achieved indeed the best BLEU score in the Arabic ASR output task, and the third best score in the Arabic Clean task, with a little more than a point BLEU difference from the best system. However, there was obviously a problem in translating from Chinese, since our system obtained nearly 11 BLEU points less than the best system. We think that our processing of the Chinese language was not adequate, and we are also investigating other possible causes.

6. Conclusions and Further work

In this year's evaluation we optimized Giza++ smoothing parameters by means of a minimum error training procedure. Alignment parameters were adjusted directly in function of automated translation metrics scores. During this procedure, only the basic n-gram MT system, with only the translation model, was used. In future work, we could consider using

[4] http://www-nlp.stanford.edu/software/lex-parser.shtml

[5] http://snowball.tartarus.org/

[6] For the system with SMR reordering the best combination was the growing forward diagonal.

	dev (dev4)	test (dev5)				
	$\frac{1}{2}$(BLEU+METEOR)	BLEU	NIST	METEOR	WER	PER
Chinese→English						
baseline	0.340	0.186	5.84	0.487	68.6	54.9
giza++ MET	0.349	0.190	5.97	0.490	69.1	54.8
giza++ MET+NNLM	0.350	0.205	6.06	0.496	69.2	54.9

Table 3: *Internal translation results for IWSLT 2007 Chinese-English task. MET refers to alignment tuning with Minimum (translation) Error Training. NNLM refers to rescoring a translation N-best list with a continuous space target language model.*

various SMT features (as would be required for a phrase-based SMT system).

In this evaluation, we have also shown the use of a neural network LM that performs probability estimation in a continuous space in the Ngram-based system. Since the resulting probability functions are smooth functions of the word representation, better generalization to unknown n-grams can be expected. The NNLM has been used to rescore the n-best lists of the Ngram-based SMT system that has participated in the 2007 IWSLT evaluation.

Our system achieved excellent scores compared to other systems in the Arabic-English task. However, it was not very competitive in the Chinese-English task. We are currently investigating the reasons for this performance difference between the two tasks.

7. Acknowledgments

This work has been partially 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) and by the Spanish Government under grant TEC2006-13964-C03 (AVIVAVOZ project).

8. References

[1] E. Vidal, "Finite-state speech-to-speech translation," *Proc. Int. Conf. on Acoustics, Speech and Signal Processing*, pp. 111–114, April 1997.

[2] A. de Gispert and J. Mariño, "Using X-grams for speech-to-speech translation," *Proc. of the 7th Int. Conf. on Spoken Language Processing, ICSLP'02*, September 2002.

[3] M. Eck and C. Hori, "Overview of the IWSLT 2005 Evaluation Campaign," pp. 11–32, October 2005.

[4] M. Paul, "Overview of the IWSLT 2006 Evaluation Campaign," in *Proc. of the Int. Workshop on Spoken Language Translation, IWSLT'06*, Kyoto, Japan, 2006, pp. 1–15.

[5] P. Lambert, R. E. Banchs, and J. M. Crego, "Discriminative alignment training without annotated data for machine translation," in *Proc. of the Human Language Technology Conference of the NAACL*, Rochester, NY, USA, 2007, pp. 85–88.

[6] H. Schwenk, M. Costa-jussà, and J. Fonollosa, "Continuous space language models for the iwslt 2006 task," in *International Workshop on Spoken Language Translation (IWSLT)*, p. 166.

[7] J. B. Mariño, R. E. Banchs, J. M. Crego, A. de Gispert, P. Lambert, J. A. Fonollosa, and M. R. Costa-jussà, "N-gram based machine translation," *Computational Linguistics*, vol. 32, no. 4, pp. 527–549, 2006.

[8] J. Mariño, R. Banchs, J. Crego, A. de Gispert, P. Lambert, J. Fonollosa, M. Costa-jussà, and M. Khalilov, "UPC's bilingual n-gram translation system," in *Proc. of the TC-STAR Workshop on Speech-to-Speech Translation*, Barcelona, Spain, June 2006, pp. 43–48.

[9] J. M. Crego, J. Mariño, and A. de Gispert, "Finite-state-based and phrase-based statistical machine translation," *Proc. of the 8th Int. Conf. on Spoken Language Processing, ICSLP'04*, pp. 37–40, October 2004.

[10] J. Crego, J. Mariño, and A. de Gispert, "Reordered search and tuple unfolding for ngram-based SMT," in *Proc. of Machine Translation Summit X*, Phuket, Thailand, September 2005, pp. 283–89.

[11] J. M. Crego, J. Mariño, and A. de Gispert, "A ngram-based statistical machine translation decoder," in *Proc. of the 9th European Conf. on Speech Communication and Technology (Interspeech)*, Lisbon, Portugal, 2005, pp. 3185–88.

[12] J. C. Spall, "Multivariate stochastic approximation using a simultaneous perturbation gradient approximation," *IEEE Trans. Automat. Control*, vol. 37, pp. 332–341, 1992.

[13] P. Lambert and R. E. Banchs, "Tuning machine translation parameters with SPSA," in *Proc. of the Int. Workshop on Spoken Language Translation, IWSLT'06*, Kyoto, Japan, 2006, pp. 190–196.

[14] N. Bertoldi, "Minimum error training (updates)," Slides of the JHU Summer Workshop (http://www.statmt.org/jhuws), 2006.

[15] J. Crego and J. Mariño, "Reordering experiments for n-gram-based SMT," in *1st IEEE/ACL Workshop on Spoken Language Technology*, Palm Beach, Aruba, December 2006, pp. 242–245.

[16] N. Habash and F. Sadat, "Arabic preprocessing schemes for statistical machine translation," in *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*. New York City, USA: Association for Computational Linguistics, June 2006, pp. 49–52. [Online]. Available: http://www.aclweb.org/anthology/N/N06/N06-2013

[17] H. Zhang, H. Yu, D. Xiong, and Q. Liu, "Hhmm-based chinese lexical analyzer ictclas," in *Proc. of the 2nd SIGHAN Workshop on Chinese language processing*, Sapporo, Japan, 2003, pp. 184–187.

[18] T. Brants, "TnT – a statistical part-of-speech tagger," in *Proc. of the Sixth Applied Natural Language Processing (ANLP-2000)*, Seattle, WA, 2000. [Online]. Available: http://www.coli.uni-sb.de/~thorsten/tnt

[19] F. J. Och, "An efficient method for determining bilingual word classes," in *Proc. of the 9th Conference of the European Chapter of the ACL (EACL)*. Association for Computational Linguistics, 1999, pp. 71–76.

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