A state-of-the-art Statistical Machine Translation System based on Moses

D. Déchelotte, H. Schwenk, H. Bonneau-Maynard, A. Allauzen and G. Adda

LIMSI/CNRS, Bât. 508, BP 133
91403 Orsay, France
dechelot,schwenk,hbm,allauzen,gadda@limsi.fr

Abstract

This paper describes a statistical machine translation system based on freely available programs such as Moses. Several new features were added, in particular a two-pass decoding strategy using n-best lists and a continuous space language model that aims at taking better advantage of the limited training data. We also investigated lexical disambiguation methods in the translation model based on POS information. The task considered in this work is the translation of the European Parliament Plenary Sessions between English and Spanish, in the framework of the TC-STAR project. The described systems performed very well in the 2007 TC-STAR evaluation.

Introduction

Automatic machine translation was one of the first natural language processing applications investigated in computer science. From the pioneer works to today's research, many paradigms have been explored, for instance rule-based, example-based, knowledge-based and statistical approaches to machine translation. Statistical machine translation (SMT) seems today to be the preferred approach of many industrial and academic research laboratories, each of them developing their own set of tools. In 1999 however, a summer workshop at Johns-Hopkins University hosted the creation of the EGYPT toolkit[1], on which the widely used training tool Giza++ (Och and Ney, 2003) is based. Later, the Pharaoh phrase-based decoder (Koehn, 2004) became available and distributed in binary form[2], but as far as we know, Pharaoh was not widely used. More recently, another workshop[3] released an open source toolkit, which includes a decoder, Moses (Koehn and al., 2007), and a comprehensive set of softwares and scripts to build a complete SMT system—namely determining word alignments, extracting phrases, performing the translation and tuning system parameters.

In this paper, we describe the development of a state-of-the-art SMT system based on the Moses suite. Several new features were added, in particular a two-pass decoding strategy using n-best lists and a continuous space language model (CSLM) that aims at taking better advantage of the limited training data. The described system participated in the 2007 TC-STAR evaluation and achieved very good rankings. We also investigated lexical disambiguation methods based on POS information, which can be interpreted as an intermediate step between "standard" phrase-based models and factored translation models. The latter approach is meant to tightly integrate linguistic information into the translation model, and is implemented in Moses, but to the best of our knowledge, experimental results have not yet been published.

This paper is organized as follows. In the next section, the LIMSI SMT system architecture is presented, as well as its training and tuning procedures, and its unique features. The following section describes the task on which the system is evaluated and the data available to train the models. Finally, experimental results are provided and commented. The paper concludes with a discussion of future research issues.

[1] http://www.clsp.jhu.edu/ws99/projects/mt/toolkit/
[2] http://www.isi.edu/licensed-sw/pharaoh/
[3] http://www.clsp.jhu.edu/ws2006/

System architecture

The goal of statistical machine translation is to produce a target sentence $\mathbf{e}$ from a source sentence $\mathbf{f}$ that maximizes the posterior probability:

$$\begin{aligned}\mathbf{e}^* &= \operatorname*{argmax}_{\mathbf{e}} \Pr(\mathbf{e}|\mathbf{f}) \\ &= \operatorname*{argmax}_{\mathbf{e}} \sum_{\mathcal{A}} \Pr(\mathbf{e}, \mathcal{A}|\mathbf{f}) \quad (1) \\ &\approx \operatorname*{argmax}_{\mathbf{e}} \max_{\mathcal{A}} \Pr(\mathbf{e}, \mathcal{A}|\mathbf{f}) \quad (2)\end{aligned}$$

In the above equations, $\mathcal{A}$ denotes a correspondence between source and target words and is called an *alignment*. The Moses decoder makes the so-called *maximum approximation* as in Equation 2.

The $\Pr(\mathbf{e}, \mathcal{A}|\mathbf{f})$ probability is modeled by a combination of feature functions, according to the maximum entropy framework (Berger et al., 1996):

$$\Pr(\mathbf{e}, \mathcal{A}|\mathbf{f}) \propto \exp \sum_i \lambda_i f_i(\mathbf{e}, \mathcal{A}|\mathbf{f}) \quad (3)$$

The translation process involves segmenting the source sentence into source phrases $\tilde{f}$; translating each source phrase into a target phrase $\tilde{e}$, and optionally reordering the target phrases to produce the target sentence $\mathbf{e}^*$. A phrase is here defined as a group of words that should be translated together (Koehn et al., 2003; Och and Ney, 2003). The segmentation stage is not modeled explicitly by any feature function, which amounts to considering every segmentation equally likely. A phrase table provides several scores that quantitize the relevance of translating $\tilde{f}$ by $\tilde{e}$. A distortion model, a language model (LM) and a word penalty are also included for a total of eight feature functions.

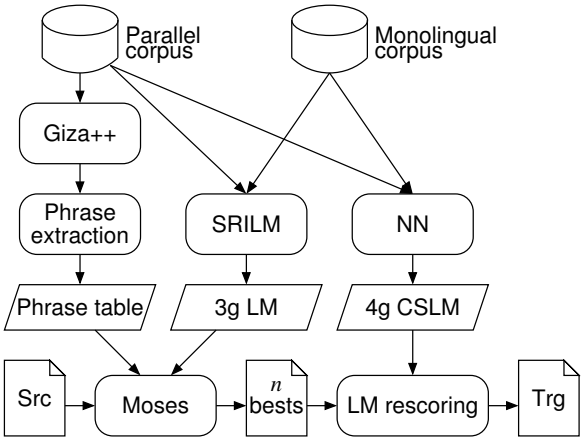

Figure 1: The LIMSI SMT system architecture

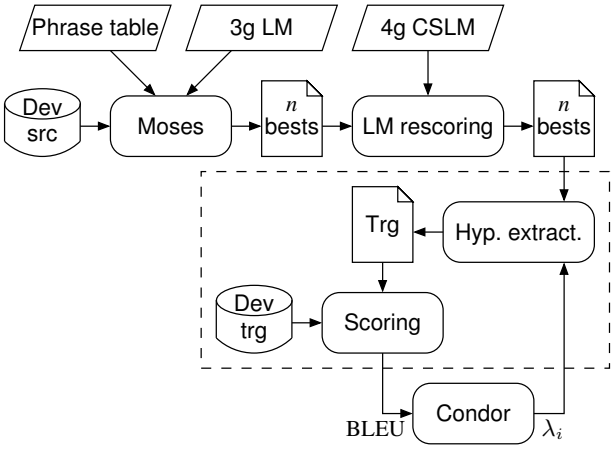

Figure 2: Tuning of second pass parameters with Condor. The dashed box denotes the "black box", needed by Condor, that outputs a BLEU score for a given set of parameters.

Phrase table acquisition

Target-to-source and source-to-target word alignments are first built with Giza++. The intersection of these two alignments is computed and augmented by word alignments present in their union, similarly to the "diag-and" algorithm from (Koehn et al., 2003). Phrase pairs consistent with the obtained alignment are extracted and scored with two relative frequency scores, two lexical scores, and one constant score, that serves as a phrase penalty (or bonus) during decoding.

Translation process

The translation process employs a two-pass strategy and is summarized in Figure 1. In the first pass, Moses generates n-best lists—1000 distinct hypotheses are requested—with a standard 3-gram language model and provides eight partial scores for each hypothesis. In the second pass, the n-best lists are rescored with a 4-gram continuous space language model and the final hypothesis is then extracted.

Parameter tuning

Each of the two passes uses its own set of eight weights and is tuned separately, a feature shared with other systems, for instance (Lööf et al., 2006; Cettolo et al., 2005). The second pass is often taken as an opportunity to compute several feature functions on the n-best list, yet after several experiments we chose not to follow this direction. The described system is thus voluntarily simple, with the hope that it will generalize well to new data. We believe that adding many feature functions, especially some that could just be ad hoc fixes to phenomena from the development data, in conjunction with performing a numerical optimization of the λ_i that is unaware of the highly discontinuous nature of BLEU (Papineni et al., 2002), bear the risk of heavily over-fitting the development data. Some experimental evidence for this are provided in the results section.

We use MERT, which is distributed along with the Moses decoder, to tune the first pass. The weights were adjusted to maximize BLEU on the development data after the first pass only. In this phase, a dozen Moses runs are necessary for each MERT optimization, and several optimization runs were started and compared during the system's development.

Tuning for the second pass is performed by Condor (Berghen and Bersini, 2005), which implements an extension of Powell's UOBYQA algorithm (Powell, 2002), and is depicted in Figure 2. The tuning procedure is as follows:

0. Using tuned first-pass weights, n-best lists are generated by Moses. These n-best lists are then rescored with the continuous space language model.

1. The n-best lists are reranked using the current set of weights. The current hypothesis is extracted and scored against the reference translations.

2. The obtained BLEU score is passed to *Condor*, which either computes a new set of weights (the algorithm then proceeds to step 1) or detects that a local maximum has been reached and the algorithm stops iterating.

The solution is usually found after about 100 iterations. It is stressed that Moses in only run once and that the whole second pass tuning operates on n-best lists.

Continuous space language model

Overall, there are roughly 60 million words of texts available to train the target language models. This is a quite limited amount in comparison to tasks like the NIST machine translation evaluations for which several billion words of newspaper texts are available. Therefore, specific techniques must be deployed to make the most of the limited resources.

In this paper, we propose to use the so-called continuous space language model. The basic idea of this approach is to project the word indices onto a continuous space and to

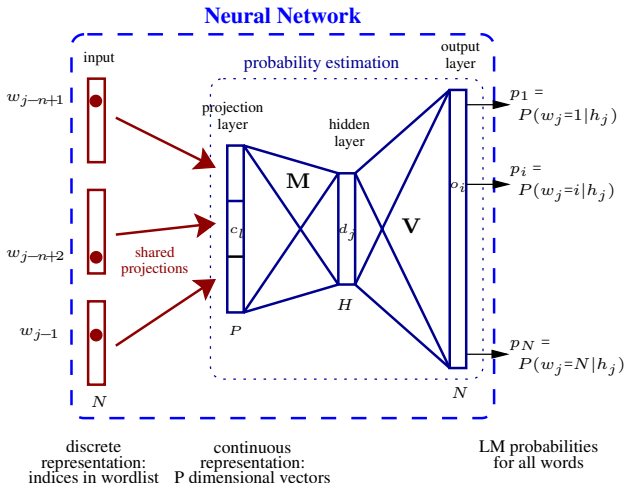

Figure 3: Architecture of the continuous space language model. h_j denotes the context $w_{j-n+1}, \ldots, w_{j-1}$. P is the size of one projection and H (resp. N) is the size of the hidden layer (resp. output layer). When short-lists are used the size of the output layer is much smaller then the size of the vocabulary.

use a probability estimator operating on this space (Bengio et al., 2003). Since the resulting probability functions are smooth functions of the word representation, better generalization to unknown n-grams can be expected. 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 language model posterior probabilities are "interpolated" for any possible context of length $n-1$ instead of backing-off to shorter contexts.

This approach was successfully applied in large vocabulary continuous speech recognition (Schwenk, 2007) and in a state-of-the-art phrase-based system for a small-domain, tourism related task (Schwenk et al., 2006). It is here applied to a broad-domain translation task.

The architecture of the neural network language model is shown in Figure 3. A standard fully-connected multi-layer perceptron is used. The inputs to the neural network are the indices of the $n-1$ previous words in the vocabulary $h_j = w_{j-n+1}, \ldots, w_{j-2}, w_{j-1}$ and the outputs are the posterior probabilities of *all* words of the vocabulary:

$$P(w_j = i|h_j) \qquad \forall i \in [1, N] \tag{4}$$

where N is the size of the vocabulary. The input uses the so-called 1-of-n coding, i.e., the ith word of the vocabulary is coded by setting the ith element of the vector to 1 and all the other elements to 0. The ith line of the $N \times P$ dimensional projection matrix corresponds to the continuous representation of the ith word. Let us denote c_l these projections, d_j the hidden layer activities, o_i the outputs, p_i their softmax normalization, and m_{jl}, b_j, v_{ij} and k_i the hidden and output layer weights and the corresponding biases. Using these notations, the neural network performs the following operations:

$$d_j = \tanh\left(\sum_l m_{jl}\, c_l + b_j\right) \tag{5}$$

$$o_i = \sum_j v_{ij}\, d_j + k_i \tag{6}$$

$$p_i = e^{o_i} \Big/ \sum_{r=1}^{N} e^{o_r} \tag{7}$$

The value of the output neuron p_i corresponds directly to the probability $P(w_j = i|h_j)$.

Training is performed with the standard back-propagation algorithm minimizing the following error function:

$$E = \sum_{i=1}^{N} t_i \log p_i + \beta\left(\sum_{jl} m_{jl}^2 + \sum_{ij} v_{ij}^2\right) \tag{8}$$

where t_i denotes the desired output, i.e., the probability should be 1.0 for the next word in the training sentence and 0.0 for all the other ones. The first part of this equation is the cross-entropy between the output and the target probability distributions, and the second part is a regularization term that aims to prevent the neural network from over-fitting the training data (weight decay). The parameter β has to be determined experimentally. Training is done using a re-sampling algorithm as described in (Schwenk, 2007).

It can be shown that the outputs of a neural network trained in this manner converge to the posterior probabilities. Therefore, the neural network directly minimizes the perplexity on the training data. Note also that the gradient is back-propagated through the projection-layer, which means that the neural network learns the projection of the words onto the continuous space that is best for the probability estimation task.

In general, the complexity to calculate one probability with this basic version of the neural network language model is dominated by the dimension of the output layer since the size of the vocabulary (up to 100k) is usually much larger than the dimension of the hidden layer (500). Therefore, the output was limited to a *short-list* composed of the s most frequent words, the other language model predictions being performed by a back-off language model. Note that this affects only the output of the neural network, all the words of the word list are considered at the network input.

Enrichment with syntactical information

It is well-known that syntactic structures vary greatly across languages. Spanish, for example, can be considered as a highly inflectional language, whereas inflection plays only a marginal role in English.

Part-of-speech (POS) language models can be used to rerank translation hypotheses, but this requires tagging the n-best lists generated by the SMT system. This can be difficult since POS taggers are not trained for ill-formed or

English: I$_{PP}$ declare$_{VVP}$ resumed$_{VVD}$ the$_{DT}$ session$_{NN}$ of$_{IN}$ the$_{DT}$ European$_{NP}$ Parliament$_{NP}$

Spanish: declaro$_{VLfin}$ reanudado$_{VLadj}$ el$_{ART}$ período$_{NC}$ de$_{PREP}$ sesiones$_{NC}$ del$_{PDEL}$ Parlamento$_{NC}$ Europeo$_{ADJ}$

Figure 4: Example of POS-tag enriched bi-text used to train the translation models

incorrect sentences. Finding a method in which morpho-syntactic information is used directly in the translation model could help overcome this drawback but also account for the syntactic specificities of both source and target languages.

Therefore, we investigate a translation model which enriches every word with its syntactic category, resulting in a sort of word disambiguation. The *enriched translation units* are a combination of the original word and the POS tag, as shown in Figure 4. The translation system takes a sequence of enriched units as inputs and outputs. This implies that the test data must be POS tagged before translation. Likewise, the POS tags in the enriched output are removed at the end of the process to provide the final translation hypothesis.

This approach also gives the flexibility to rescore the n-best lists using either a word language model, a POS language model or a language model of enriched units.

POS tagging was performed with the *TreeTagger* (Schmid, 1994). This software provides resources for both of the considered languages and it is freely available. *TreeTagger* is a Markovian tagger that uses decision trees to estimate trigram transition probabilities. The English version is trained on the *PENN treebank* corpus[4] and the Spanish version on the *CRATER* corpus.[5]

Tasks and data

The task considered in this work is the translation of the European Parliament Plenary Sessions (EPPS) between English and Spanish. The following experiments were carried out in the framework of the TC-STAR project, which is envisaged as a long-term effort to advance research in all core technologies for speech-to-speech translation. The training material consists of the minutes edited by the European Parliament in several languages, known as the Final Text Editions (Gollan et al., 2005). These texts were aligned at the sentence level and are used to train the statistical translation models (see Table 1 for some statistics).

Three different conditions are considered in the TC-STAR evaluation: translation of the Final Text Edition (*text*), translation of the transcriptions of the acoustic development data (*verbatim*) and translation of speech recognizer output (*ASR*). Here we only consider the *verbatim* condition, translating between English and Spanish (both ways). Specifics of translating automatic speech transcriptions are described in (Déchelotte et al., 2007). For the *verbatim* task, the development and test data consists of about 30k words. The test data is partially collected in the Spanish Parliament,

		Spanish	English
Whole parallel corpus	Sentence Pairs	1.2M	
	Total # Words	34.1M	32.7M
	Vocabulary size	129k	74k
Sentences shorter than 40 words	Sentence Pairs	0.91M	
	Total # Words	18.5M	18.0M
	Word vocabulary	104k	71k

Table 1: Statistics of the parallel texts used to train the statistical machine translation system.

Corpus	English	Spanish
EPPS	36.5	37.8
Audio transcriptions	1.6	0.8
Hansard/Cortes	57.7	50.4

Table 2: Number of words (in millions) of the various mono-lingual corpora.

which results in a small mismatch between the training and the test data. Scoring is case sensitive, includes punctuation marks and uses two reference translations.[6]

Additional mono-lingual data is used to train the target language models. Audio transcriptions of European parliament sessions are available in English and Spanish, and transcriptions from the Spanish parliament are available as well. A third and last mono-lingual data source was the British Hansard corpus (proceedings of the British parliament) and the Cortes corpus (proceedings of the Spanish parliament). Word counts are collected in Table 2.

Parallel corpus filtering

The distributed corpus includes some meta-information, for instance speaker names, and session dates and topics. This information was discarded by stripping out all lines containing an angle bracket (< or >). The parallel data occasionally contains sentences with no translations, which have to be deleted from the translation model training set.

Additionally, sentence pairs that did not seem to be actually translations of each other were removed. The pruning criteria only relies on the number of characters of the source sentence and its alleged translation. Specifically, if one of the sentences is less than 10 times shorter than its

[4] http://www.cis.upenn.edu/~treebank

[5] http://www.comp.lancs.ac.uk/linguistics/crater/corpus.html

[6] See http://www.elda.org/en/proj/tcstar-wp4/ for details on the specifications and the available training data.

counterpart, or less than 4 times shorter but longer than 20 characters, the sentence pair is dismissed.

The parallel texts were converted from UTF-8 to Latin1 encoding. This process highlighted a great number of spurious symbols, some of which could be semi-automatically corrected. For example, the character ń occurred 67 times and ought to be replaced by the character ñ, and all 15 occurrences of the character σ found in the original data were actually meant to be ó.

As English and Spanish words can be adequately represented in Latin1 encoding, the only "valid" uses of UTF-8 stem from proper names, which had to be artificially converted to Latin1 and for which the correct UTF-8 form is restored after translation. This process was again the opportunity to detect and fix normalization issues, as proper nouns containing non-latin characters were frequently mistyped. As an example, the correct spelling of `Milošević` appears only three times in the corpus, whereas various spellings with one or zero accentuated characters appear several hundred times.

Language dependent preprocessing

The translation model is trained on parallel texts extracted from the European Parliament website[7], whose format differ from the "verbatim" format in several ways:

Original training data	Verbatim condition
Case follows common orthographic conventions, the first word in the sentence is capitalized.	True case: the first word in the sentence is not capitalized, unless it corresponds to its normal spelling.
Some punctuation marks next to words without space, as normal.	Punctuation marks separated from words.
Number, dates, and other quantities in digits or abbreviated.	Number, dates, and other quantities explicited in words.
Text edited for fluency.	Speech transcription, with disfluencies.

Some of the work of adapting the training data to the evaluation condition had already been done in-house for the Speech-To-Text (STT) task, yet several aspects had to be modified specifically for the machine translation task. This includes a few "obvious" normalizations, e.g. for some key words (`Mister` or `Mr.`, `señor` or `Sr.`, etc) and acronyms (as some acronyms are spelled out for the speech recognition task).

In English, several criteria to split words at hyphens have been compared. An algorithm that relies on a word list and splits compounds very conservatively was found to outperform the "baseline" word splitting algorithm deployed in our STT system by roughly 0.5 BLEU, as measured in the early stages of the translation system development. For example, numbers like `forty-two` and compounds like `pro-European` may be split for the STT task but are better translated when left in one token.

[7] http://www.europarl.europa.eu/

Experimental results

In this section, detailed results of the different variants of our system are provided. Design decisions and parameter tuning were performed on the development data (Dev06), and the generalization behavior was estimated on the test data of last year's TC-STAR evaluation (Test06). For this, the system was run with exactly the same parameters than those used on the development data, without further tuning. The systems also participated in the official evaluation organized by the TC-STAR consortium in February 2007. Results of our systems are provided here for completeness (Test07).

Performance of the target language models

The first pass of the translation process (n-best list generation with Moses) makes use of 3-gram back-off language models. The models for English were trained on the EPPS data, the transcriptions of the audio data and the Hansard corpus. The models for Spanish were trained on the EPPS data, the transcriptions of the European and Spanish parliament audio data and the proceedings of the Spanish Parliament. For each language, three independent language models were first built on each corpus and then linearly interpolated so as to minimize perplexity on the development data.

The continuous space language model was trained on exactly the same data. The shortlist length was set to 8k for both languages. The continuous space language model is not used alone but interpolated with several n-gram models. First of all, the neural network and the reference back-off models are interpolated together—this always improved performance since both seem to be complementary. Second, several neural networks with different sizes of the continuous representation were trained and interpolated together. This usually achieves better generalization behavior than training one larger neural network. The interpolation coefficients were calculated by optimizing perplexity on the development data, using an EM procedure. The obtained values are about 0.3 for the back-off language model, the rest being roughly equally distributed over the continuous space language models. This interpolation is used in all our experiments. For the sake of simplicity we will still call

Language	Back-off LM		CSLM
	3-gram	4-gram	4-gram
English	134.5	123.4	102.7
Spanish	70.3	64.0	54.5

Table 3: Perplexities on the development data for back-off and continuous space language models (CSLM).

Translation direction	Translation units	Dev06			Eval06			Eval07		
		3g	4g	CSLM	3g	4g	CSLM	3g	4g	CSLM
Spanish→English	words	47.20	47.64	48.26	50.96	51.23	51.66	48.42	48.67	49.19
English→Spanish	words	48.78	49.39	50.15	48.38	49.06	50.20	49.19	50.17	51.04
	enriched	48.92	49.45	50.30	48.71	49.00	49.96	49.13	49.91	51.04

Table 4: BLEU scores on the development and test data. CSLM denotes the continuous space language model.

this the continuous space language model. Table 3 summarizes the perplexities of all the language models used in our system.

Translation performance

Table 4 gives a result summary of the developed systems for both translation directions. When translating from Spanish to English, the BLEU score increases by about 0.4 on the development data and 0.3 on the test data when rescoring the n-best lists with a 4-gram language model. The continuous space language model achieves an additional improvement of 0.6 BLEU on the development data and up to 0.5 on the test data. Good language models seem to be more important when translating to Spanish. The use of a 4-gram gives an 0.6 improvement and the continuous space language model brings another 0.8 BLEU. Our systems also exhibits a very good generalization behavior: the improvements obtained on the test data are as good or even exceed those observed on the development data. The importance of the Spanish language model when translating from English can be explained by the additional inflections present in Spanish: the target language model is then crucial to select the correct inflected forms.

Table 5 provides additional automatic evaluation metrics, as well as the result range for all systems in the 2007 TC-STAR evaluation. The systems described in this paper ranked first in the English to Spanish translation task and in third position when translating from Spanish to English. Interestingly, our systems achieved poorer rankings on the development data (results not detailed here). This can be seen as experimental evidence that "simple" systems may generalize better than systems with many feature functions.

	BLEU	NIST	mWER	mPER
S→E	49.19 (42.95–49.60)	10.67 (9.81–10.83)	39.8% (39.7–44.9)	27.4% (27.4–31.7)
E→S	51.04 (37.39–51.04)	10.29 (8.38–10.34)	37.9% (37.1–51.4)	28.8% (28.8–38.3)

Table 5: Automatic evaluation metrics of the 2007 TC-STAR evaluation of our system and the result range from all the participants (in parentheses).

Minor improvements in the BLEU scores were obtained using much larger n-best lists (up to 10,000 were tried), but at the expense of a prohibitive processing time.

Lexical disambiguation

Lexical disambiguation based on POS information has only been applied when translating from English to Spanish (Bonneau Maynard et al., 2007). The results are summarized in the last line of Table 4. Although small improvements may be observed on the development data, they do not carry over to the test data. Still, it can be noticed that the enriched unit system always outperforms the baseline word system after the first pass; but there is no significant difference after rescoring the n-best lists with the continuous space language model. We conjecture that both approaches correct the same translation problems.

The results reported in Table 4 were obtained by rescoring with word language models, even in the last row. We believe that it is necessary to use the enriched representation also in the language models in order to take full advantage of the disambiguation in the translation model. Rescoring with simple POS language models was tried, but without success. We are now working on the use of factored language models (Bilmes and Kirchhoff, 2003) that simultaneously use the word and POS information.

Figure 5 shows comparative translation examples from the baseline and the enriched translation systems. In the first example, the baseline system outputs "*durante los últimos sesiones*" where the enriched translation system produces "*en los últimos períodos de sesiones*", a better translation that may be attributed to the introduction of the masculine word "*períodos*", allowing the system to build a syntactically correct sentence. In the second example, the syntactical error "*no puede ser un cierto reconocimiento*" produced by the baseline system induces an incorrect meaning of the sentence, whereas the enriched translation system hypothesis "*existe un cierto reconocimiento*" is both syntactically and semantically correct. These examples could be seen as experimental evidence that lexical disambiguation seems to improve the translation quality although this is not necessarily measured by the BLEU score.

Conclusion

This paper described a statistical machine translation system based on freely available programs such as Moses. The task considered is the translation of the European Parliament Plenary Sessions between English and Spanish, in the framework of the TC-STAR project. A two-pass decoding strategy was described, which enabled the use of a continuous-space language model in order to take better advantage of the limited amount of in-domain language model

English :	you will be aware President that over the last few sessions in Strasbourg. ..
Baseline:	usted sabe que el Presidente *durante los últimos sesiones* en Estrasburgo ...
Enriched units:	usted sabe que el Presidente *en los últimos períodos de sesiones* en Estrasburgo ...
English :	... in this house there might be some recognition ...
Baseline:	... en esta asamblea *no puede ser un cierto reconocimiento* ...
Enriched units:	... en esta asamblea *existe un cierto reconocimiento* ...

Figure 5: Comparative translations using the baseline word system and the enriched unit system.

training data. The described system is voluntarily "simple", in that it only uses eight feature functions. This contrasts with an apparent tendency in the literature to use many feature functions, each one obtaining a small improvement on the development data. Based on the limited experiments described in this paper, a "simple" system may generalize better on the test data: all of the systems achieved very good results in the 2007 TC-STAR evaluation. We also described work on lexical disambiguation in the translation model using POS information, but were unable to obtain significant improvements on the test data with this technique. We plan to pursue this direction by using factored representations in the translation and language model.

Acknowledgments

This work has been partially funded by the European Union under the integrated project TC-STAR (IST-2002-FP6-506738), and by the French Government under the project INSTAR (ANR JCJC06_143038).

References

Yoshua Bengio, Rejean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of Machine Learning Research*, 3(2):1137–1155.

Adam L. Berger, Vincent J. Della Pietra, and Stephen A. Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71.

Frank Vanden Berghen and Hugues Bersini. 2005. CONDOR, a new parallel, constrained extension of Powell's UOBYQA algorithm: Experimental results and comparison with the DFO algorithm. *Journal of Computational and Applied Mathematics*, 181:157–175.

Jeff A. Bilmes and Katrin Kirchhoff. 2003. Factored language models and generalized parallel backoff. In *Proc. of NAACL 2003*, pages 4–6, Edmonton, Canada.

Hélène Bonneau Maynard, Alexandre Allauzen, Daniel Déchelotte, and Holger Schwenk. 2007. Combining morphosyntactic enriched representation with n-best reranking in statistical translation. In *NAACL-HLT Workshop on Syntax and Structure in Statistical Translation*, pages 65–71, Rochester, New York, April.

M. Cettolo, M. Federico, N. Bertoldi, R. Cattoni, and B. Chen. 2005. A look inside the ITC-irst SMT system. In *Proc. of the tenth MT Summit*, pages 451–457, Phuket, Thailand, September.

Daniel Déchelotte, Holger Schwenk, Gilles Adda, and Jean-Luc Gauvain. 2007. Improved machine translation of text-to-speech outputs. In *Proc. of InterSpeech*.

Christian Gollan, Maximilian Bisani, Stephan Kanthak, Ralf Schlüter, and Hermann Ney. 2005. Cross domain automatic transcription on the TC-STAR EPPS corpus. In *Proc. of ICASSP 2005*, Philadelphia, USA, March.

Philipp Koehn and al. 2007. Moses: Open source toolkit for statistical machine translation. In *ACL, demonstration session*, Prague, Czech Republic, June.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proc. of the Human Language Technology Conference 2003 (HLT-NAACL 2003)*, Edmonton, Canada, May.

Philipp Koehn. 2004. Pharaoh: a beam search decoder for phrase-based statistical machine translation models. In *AMTA 2004*, Washington, USA.

J. Lööf, M. Bisani, Ch. Gollan, G. Heigold, B. Hoffmeister, Ch. Plahl, R. Schlüter, and H. Ney. 2006. The 2006 RWTH parliamentary speeches transcription system. In *Proc. of ICSLP*, pages 105–108, Pittsburgh, USA, September.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proc. of the 40th Annual Meeting of the ACL*, University of Pennsylvania.

M.J.D. Powell. 2002. UOBYQA: Unconstrained optimization by quadratic approximation. *Mathematical Programming*, 92(3):555–582.

Helmut Schmid. 1994. Probabilistic part-of-speech tagging using decision trees. In *Proc. of International Conference on New Methods in Language Processing*, Manchester, UK, September.

Holger Schwenk, Marta R. Costa-Jussà, and José A. R. Fonollosa. 2006. Continuous space language models for the IWSLT 2006 task. In *Proc. of IWSLT 2006*, pages 166–173, Kyoto, Japan, November.

Holger Schwenk. 2007. Continuous space language models. *Computer Speech and Language*, 21:492–518.