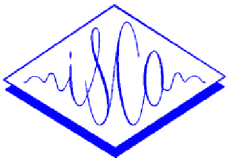
Proceedings of IWSLT 2009, Tokyo - Japan



UPV Translation System for IWSLT 2009

Guillem Gascó , Joan Andreu Sánchez

Instituto Tecnológico de Informática
Universidad Politécnica de Valencia, Spain
ggasco@iti.upv.es , jandreu@dsic.upv.es

Abstract

In this paper, we describe the machine translation system developed at the Polytechnic University of Valencia, which was used in our participation in the International Workshop on Spoken Language Translation (IWSLT) 2009. We have taken part only in the Chinese-English BTEC Task. In the evaluation campaign, we focused on the use of our hybrid translation system over the provided corpus and less effort was devoted to the use of pre- and post-processing techniques that could have improved the results.

Our decoder is a hybrid machine translation system that combines phrase-based models together with syntax-based translation models. The syntactic formalism that underlies the whole decoding process is a Chomsky Normal Form Stochastic Inversion Transduction Grammar (SITG) with phrasal productions and a log-linear combination of probability models. The decoding algorithm is a CYK-like algorithm that combines the translated phrases inversely or directly in order to get a complete translation of the input sentence.

1. Introduction

The Phrase-based Machine Translation (PBT) approach has been demonstrated to be one of the best approaches in translation of similarly structured language pairs (Spanish-English, French-Italian...) [1]. The main advantages of PBT systems are that the phrase pairs can be extracted easily from a word alignment and the process of decoding is usually fast and easy. In addition, phrase models usually have high sentence coverage.

One of the weaknesses of the phrase-based models is the problems they have incorporating syntactic information in the translation. For instance, most English sentences contain a subject and a verb, but there is no way of including this information in a traditional phrase-based system. Syntactic motivated reorderings are also very difficult to include in phrase-based systems.

Another common approach within the field of SMT is the so called Syntax-based Machine Translation (SBT). This approach is characterized by the use of syntactic information in the process of MT. For pairs of languages with a high number of reorderings (Spanish-German, Chinese-English...) SBT seems to be a very interesting solution. The strength of SBT systems is that outputs tend to be syntactically well formed and the reordering can be influenced by syntactic context [2]. However, SBT systems usually have poor sentence coverage.

Hierarchical MT systems [3] combine phrases (with gaps) for translation and a syntax-based decoding algorithm. Some of them even use linguistically motivated information [4]. The result of this combination is a very large set of rules (parameters) that, in most cases, must be pruned for practical issues. The number of rules produced by a hierarchical translation system is very high, which makes the training and decoding processes slow.

For these reasons, we present a decoding system that also uses phrases for translation and a syntax-based decoding, but has easier and faster algorithms for training and decoding. The syntactic formalism that underlies the whole decoding process is a Chomsky Normal Form Stochastic Inversion Transduction Grammar (SITG) with phrasal productions and a log-linear combination of probability models instead of a single probability.

The rest of the paper is structured as follows: Section 2 presents the theoretical framework that underlies the whole translation process. Section 3 explains the translation algorithm followed by the decoder, and Section 4 gives some details of the process. Section 5 shows the training algorithms used to obtain the phrasal ITG. Section 6 present the experiments carried out, and, finally, Section 7 presents the conclusions of the paper.

2. Theoretical framework

Inversion Transduction Grammars (ITG) [5] are a special kind of Synchronous Grammar whose parse algorithms have polynomial complexity. This fact leads us to the use of ITG as the main model for the translation process. An ITG is a tuple $(\mathcal{N}, \Sigma, \Delta, S, \mathcal{R})$ where $\mathcal{N}$ is the set of non-terminals, $S \in \mathcal{N}$ is the root non-terminal, Σ is the input alphabet, Δ is the output alphabet, and $\mathcal{R}$ is a set of rules. Rules can be divided into two sets: syntactic rules and lexical rules. Syntactic Rules have the form $A \rightarrow [BC]$ or $A \rightarrow \langle BC \rangle$, where A, B, and C are non-terminals. The brackets that enclose the right part of the rule mean that the two non-terminals are expanded in the same order in the input and output lan-

guages, whereas the rules with pointed bracketing expand the left symbol into the right symbols in direct order in the input language and in reverse order in the output language. Lexical rules have the form $A \rightarrow x/y$ where $x \in \{\Sigma \cup \epsilon\}$ and $y \in \{\Delta \cup \epsilon\}$. It must be noted that x or y can be the empty string, denoted by ϵ, but not both in the same production.

There are two major problems with the use of ITG:

1. As stated by Wu, ITGs cannot represent all possible permutations of words that may occur during translation. However, in [6], the authors state that only a small percentage of human translations cannot be represented by an ITG transduction .

2. Grammars of this kind do not take into account direct multi-word (or phrasal) transduction. An ITG gets the transduction of a segment of words only by combining the transductions of its constituents.

In order to at least partially overcome such problems, we present a phrasal extension of ITG: the so-called Phrasal ITG (PhITG). PhITG has already been used in the literature [7]. PhITGs are very similar to ITG; the only difference is that the lexical rules can produce strings directly instead of a single word in each of the languages. This changes the definition of lexical rules. A lexical rule now has the form $A \rightarrow x/y$ with $x \in (\Sigma)^*$ and $y \in (\Delta)^*$. One side of the lexical production (x or y) can be the empty string, but not both. Figure 1 shows two different PhITG-parsings for a given pair of sentences in English and Spanish. In part **b)** the non-terminal NP produces a *phrase* translation.

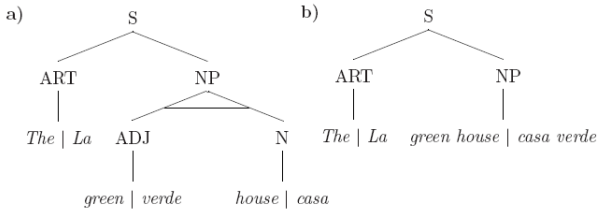


Figure 1: *Two different Phrasal ITG parsings for a pair of sentences English-Spanish.*

Formally, for any bilingual strings $u = (u^s/u^t)$, $v = (v^s/v^t)$, where $u^s, v^s \in (N \cup \Sigma)^*$ are the source part, and $u^t, v^t \in (N \cup \Delta)^*$ are the target part, we say that u directly yields v, written as $u \Rightarrow v$ if there exists a rule $(\alpha \rightarrow \beta^s/\beta^t)$ in $\mathcal{R}$ such that $u = (u_1^s \alpha u_2^s / u_1^t \alpha u_2^t)$ and $v = (v_1^s \beta^s v_2^s / v_1^t \beta^t v_2^t)$. We write $u \stackrel{r}{\Rightarrow} v$, when the application of the rule $r = \alpha \rightarrow \beta^s/\beta^t$ produces v from u.

In order to get a unified framework, we always substitute the most left non-terminal. For any u, v, $u \Rightarrow^* v$, we say that u yields v, if there exist $r_1, r_2, \ldots r_k$ such that $u \stackrel{r_1}{\Rightarrow} u_1 \stackrel{r_2}{\Rightarrow} u_2 \stackrel{r_3}{\Rightarrow} \ldots \stackrel{r_k}{\Rightarrow} v$. Each of the ordered sets of rules $\{r_1, r_2, \ldots r_k\}$ that is used to produce v from u is called derivation from u to v. If u yields v following the derivation D, we express it as $u \stackrel{D}{\Rightarrow} v$. We say that a non-terminal A yields a bitext x/y when $A \stackrel{*}{\Rightarrow} x/y$. The bilingual language generated by the PhITG is the set of bilingual sentences x/y such that $S \stackrel{*}{\Rightarrow} x/y$.

In a way similar to the way that Stochastic ITGs assign a probability to each of the rules, we can assign a loglinear combination of probability models [8] to the rules of a PhITG. The models used are:

Direct Translation Probability: Probability of the target sentence given the source sentence: $h_1 = Pr(t|s)$

Inverse Translation Probability: Probability of the source sentence given the target sentence: $h_2 = Pr(s|t)$

Lexical Direct Probability: Probability of translation of the source sentence words into the target sentence words using an IBM1 translation model: $h_3 = Pr_{IBM}(t|s)$

Lexical Inverse Probability: Probability of translation of the target sentence words into the source sentence words using an IBM1 translation model: $h_4 = Pr_L(s|t)$

N-gram Language Model: Probability of the target sentence using a n-gram language model: $h_5 = Pr_{LM}(t)$

Syntactic Probability: Probability of the rules of the derivation: $h_6 = Pr_R(D)$.

Word Penalty Factor: This feature is used to model the length of the output: $h_7 = exp(|t|)$, with $|t|$ being the number of words of the target sentence.

Phrase Penalty Factor: This feature is used to control the number of phrases used in the translation.

The probability of a derivation D is, then:

$$\Pr(D) \propto \prod_i h_i(D)^{\lambda_i} \tag{1}$$

where $h_i(D)$ is the set of features over derivations described above and λ_i are feature weights. Note that from the derivation D we can get the source language and the target language strings, s and t respectively.

Apart from the n-gram language model probability of the target language ($\Pr_{LM}(t)$), all the other models can be computed as the products of functions on the rules used in a derivation. Then,

$$h_i(D) = \prod_{(X \rightarrow (\gamma,\alpha)) \in D} h_i(X \rightarrow (\gamma, \alpha)) \tag{2}$$

If we do not use the language model score, a CYK-like algorithm can be used to obtain the best or the n-best derivations. However, the addition of the n-gram language model is problematic. This problem will be discussed in Section 4.

3. Translation algorithm

This section describes the basic algorithm for obtaining the target language sentence t that maximizes the probability of translation using the translation model described in Section 2. Suppose that the n-gram language model is not used. Now imagine that we know the source sentence s and we need to know the string in the target language that maximizes the probability of translation. Let s_i^j be the phrase that contains the source sentence words from position i to j. Then, we define

$$\delta_{ij}(A) = \max_t \Pr(A \stackrel{*}{\Rightarrow} s_i^j/t) \tag{3}$$

as the maximum probability of any parse tree that yields the bilingual string s_i^j/t from the non-terminal symbol A, where t is any target language sentence.

The recursive algorithm shown in Figure 2 can be used in order to compute $\delta_{ij}(S)$.

$$\text{For all } A \in N \text{ and } i,j \text{ such that } \begin{cases} 0 \le i < j \le |s|, \\ j - i \ge 1, \end{cases} \tag{4}$$

$$\delta_{ij}(A) = \max(\delta_{ij}^{[]}(A), \delta_{ij}^{\langle\rangle}(A), \max_t \Pr(A \to s_i^j/t)) \tag{5}$$

where

$$\delta_{ij}^{[\,]}(A) = \begin{cases} \max\limits_{\substack{B,C \in N \\ i<I\le j}} \Pr(A \to [BC])\delta_{iI}(B)\delta_{Ij}(C) & \\ & \text{if } j - i > 1 \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

$$\delta_{ij}^{\langle\rangle}(A) = \begin{cases} \max\limits_{\substack{B,C \in N \\ i<I\le j}} \Pr(A \to \langle BC\rangle)\delta_{iI}(B)\delta_{Ij}(C) & \\ & \text{if } j - i > 1 \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

Figure 2: *Algorithm for computing the maximum probability of all the derivations that yield the sentence pair s/t from the non-terminal symbol A, given s.*

Now we define

$$\tau_{ij}(A) = \operatorname*{argmax}_t(\Pr(A \Rightarrow^* s_i^j/t)) \tag{8}$$

as the target language phrase t that maximizes the probability of derivation from the non-terminal symbol A to the bilingual string (s_i^j/t). We can obtain $\tau_{ij}(A)$ by means of the algorithm shown in Figure 3. Thus, a new translation hypothesis can be created through three different processes: a new phrase pair that covers the whole source part, and the inverse or direct combination of two hypotheses previously computed. Since we are using a context-free-like model, the use of these search algorithms guarantees finding the most likely translation. A simple CYK-like bottom-up algorithm can be used to compute it.

$$\tau_{ij}(A) = \begin{cases} t & \text{if } \Pr(A \to s_i^j/t) \\ & \text{is the maximum in (5)} \\ \tau_{iI}(B)\tau_{Ij}(C) & \\ & \text{if } \Pr(A \to [BC])\delta_{iI}(B)\delta_{Ij}(C) \\ & \text{is the maximum in (5)} \\ \tau_{Ij}(C)\tau_{iI}(B) & \\ & \text{if } \Pr(A \to \langle BC\rangle)\delta_{iI}(B)\delta_{Ij}(C) \\ & \text{is the maximum in (5)} \end{cases} \tag{9}$$

Figure 3: *Algorithm for obtaining the target sentence that maximizes the probability over all the derivations given the source sentence s.*

4. Decoding details

In this section, we give some details of the decoding process, such as the inclusion of the n-gram language model and some optimization strategies.

The use of n-gram language models has been demonstrated to be very useful for PBT systems. However, in contrast to the other models, the n-gram language model probability of a derivation cannot be computed as a product of the language model probabilities of the rules used in the derivation (it depends on the context). The most likely translation of a sentence may use partial hypotheses that were not the most likely in their respective cells of the CYK chart. Hence, when including the n-gram language model, the optimality of the CYK algorithm is no longer guaranteed and using it is not enough to obtain the most likely translation.

In order to partially alleviate these problems, we need to use a translation hypotheses stack (from now on referred to as Agenda) in each cell of the CYK-like chart instead of a single hypothesis. The hypotheses of two Agendas can be combined directly or inversely, and the n-gram language model score of the new resulting hypotheses must be recomputed. Figure 4 shows an example of the combination of two different Agendas. First the decoder uses the direct combination by means of the rule SN $\to$ [ADJ NN], its language model is recomputed and the new hypotheses enter in the Agenda. The same happens with the inverse combination with the rule SN $\to \langle$ADJ NN$\rangle$.

This algorithm could be used to exhaustively search through the whole space of hypotheses. However, for big corpora and long sentences, some kind of restriction over the search space must be applied. When two hypotheses in an

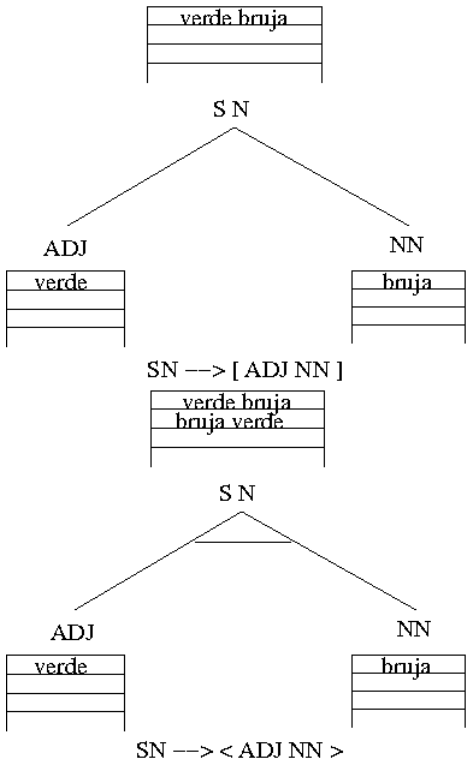


Figure 4: *Inverse and direct combination of two hypotheses.*

Agenda have the same target language part, the less likely hypothesis can be discarded without the risk of losing the optimality. We call this process hypotheses recombination.

The risk-free hypotheses recombination is usually not enough. Hence, we also use two kinds of pruning that can make the decoder lose the most likely translation:

1. Histogram Pruning: Only the n most likely hypotheses are stored in each agenda.

2. Beam Pruning: We only store a hypothesis in an agenda if its probability is greater than $\gamma \cdot \Pr(h^*)$, where h^* is the hypothesis with the highest probability and γ is a real number between 0 and 1.

Both pruning strategies are parameterizable, so there is a choice between a slow but precise search or a fast and more inaccurate one.

5. Training the system

In this section, we describe the process of obtaining a SPhITG from a corpus. In order to obtain an easy process, to avoid the sparseness of the parameters, and to make the use of phrasal tables from PBT systems possible, we make a simplification assumption: we assume that the probability of all the phrasal rules with more than one word on the source language side is the same for all the non-terminal symbols, that is:

$$\Pr(A \rightarrow x_i^j / y_k^l) = \Pr(x_i^j / y_k^l) \text{ when } j - i > 1 \qquad (10)$$

This assumption makes it possible to split the training process into two parts:

- Training the phrase table that models the phrasal productions.

- Training the SITG that models the syntactic and single word productions.

The first part is the same as the training process of a PBT system. We followed the first method explained in [9].

In order to get a SITG, we used the following method:

1. Create an initial SITG: We assigned the probability of alignment of the words of the corpus $\Pr(s|t)$ (IBM models) to the lexical rules of the form $A \rightarrow s/t$. Then, we created all the possible syntactic rules (direct and inverse) using all the non-terminal symbols and assigning a low random probability to them. The grammar was smoothed by adding all the possible rules of the form $A \rightarrow s/\epsilon$ and $A \rightarrow \epsilon/t$ with a low probability.

2. Reestimation: With the initial SITG, we applied several iterations of the Viterbi reestimation algorithm. Hence, we parsed the corpus to get the most likely parse tree for each pair of sentences of the corpus. Then we reestimated the probabilities of the productions of the SITG by counting and normalizing the occurrences of the rules in all the trees.

3. Assign linguistic information to the non-terminal symbols: We parsed the source part of the corpus with a linguistic language parser. Then we used the SITG obtained in step 2 to parse the bilingual corpus. Finally, we associated the non-terminals of the linguistic parse trees with the non-terminals of the SITG trees and reestimated the new probabilities of the rules by counting and normalizing.

6. Experiments

In this section, we describe the experiments carried out on the corpus provided. All the results were computed over the lowercased and tokenized corpus. The phrase tables were extracted using the training method of Moses software[10] and the alignments for the initial SITG were computed using GIZA++ [11]. The weights of the log-linear combination of models were computed using the Minimum Error Rate training software ZMERT [12]. The linguistic parse trees for the SITG training were computed using the Chinese part of the Stanford Parser [13]. All the configurations in these experiments used a 5-gram language model obtained from the corpus with the software SRILM [14].

6.1. Data

The experiments described in this section were carried out using the training and development sets provided for the Chinese-English BTEC task of the IWSLT evaluation campaign. There were 5 different development sets (devsets 1, 2, 3, 6, and 7). We added devsets 1, 2 and 3 to the training set, devset6 was used for tuning the system, and devset7 was used as a blind test set. The statistics of the resulting training, development and test sets are shown in Table 1. Since the development set is a multi-reference file, the number on the English side of the table is, in fact, the number of words divided by the number of references.

		Chinese	English
Training	Sentences	42,655	
	Words	330,163	380,431
	Vocabulary Size	8,773	8,387
DevSet	Sentences	489	
	Words	3,169	3,861
	OOV Words	111	115
Test	Sentences	507	
	Words	3,357	-
	OOV Words	97	-

Table 1: *Statistics for the partitions of the BTEC corpus used.*

6.2. Results

For the baseline, we used a PBT system with the same phrase table as our hybrid system. Three different configurations for the hybrid decoder were tested:

Configuration 1: The SITG used is the initial one.

Configuration 2: We reestimate the SITG using the Viterbi reestimation algorithm.

Configuration 3: Reestimated SITG with linguistic non-terminal symbols.

The results presented were evaluated with respect to the BLEU machine translation evaluation metric [15]. The results obtained using the partitions described above are reported in Table 2. These results were obtained after tokenizing and lowercasing the corpus. The SITG of Configuration 1 did not provide important syntactic information to the system, so the decoding process was almost completely driven by the phrase table and the language model. For that reason, the results of the PBT and the Configuration 1 were quite similar. The reestimation of the SITG and the association of linguistic meaning to the non-terminal symbols significantly improved the performance of the system (an improvement of 1.73 points in the %BLEU score).

Figure 5 shows the comparison between the output of the PBT decoder, the output of the hybrid decoder, and one of the references. In the first sentence, it can be seen how the hybrid system output is syntactically better formed than the PBT output sentence. The reordering of PBT systems is usually guided by the language model and sometimes by lexical reordering tables. This information is sometimes not enough. This fact can be seen in the second sentence. The PBT system changed the order of the numbers, while the hybrid system learned that numbers should not be inverted. The rules of the SITG involved in this reordering and their probabilities are shown in Figure 6. Note that the probability for direct combination is higher than for inverse combination in these rules.

System	%BLEU
Baseline (PBT)	41.1
Hybrid Decoder (Conf. 1)	41.23
Hybrid Decoder (Conf. 2)	41.79
Hybrid Decoder (Conf. 3)	42.85

Table 2: *Results of the experimentation in %BLEU score.*

$$\Pr(\text{QP} \rightarrow [\text{CD CD}]) = 0.147$$
$$\Pr(\text{QP} \rightarrow \langle\text{CD CD}\rangle) = 0.046$$
$$\Pr(\text{QP} \rightarrow [\text{CD QP}]) = 0.284$$
$$\Pr(\text{QP} \rightarrow \langle\text{QP CD}\rangle) = 0.061$$

Figure 6: *Rules involved in the reordering of numbers. QP and CN are non-terminal symbols that mean quantified phrase and cardinal number, respectively.*

Thus, we used Configuration 3 to translate the test of the evaluation campaign. The official results, in %BLEU, TER, and NIST scores are shown in Table 3.

System	%BLEU	TER	NIST
case+punc	35.29	41.86	6.04
no case+no punc	35.15	46.96	6.19

Table 3: *Official results in %BLEU, TER and NIST scores for the Chinese-English BTEC task with the evaluation specifications case+punc and no_case+no_punc.*

7. Conclusions

This paper describes the hybrid translation system presented by the Polytechnic University of Valencia for the IWSLT 2009 evaluation campaign. The hybrid decoder uses phrase tables together with SITGs for the translation. The results show that the use of syntactic information improves the performance of the system. When the system does not use syntactic information, its performance is similar to the PBT system results.

PBT translation	this one and what 's the difference between ?
Hybrid translation	what 's the difference between this with that ?
Reference	how is this one different from that one ?
PBT translation	please wait a moment . call mr. is three four one four five six seven .
Hybrid translation	please wait a moment . call mr. is three six four five seven four one .
Reference	just a moment , please . the number for s nicholas is three six four five seven four one .
PBT translation	can i go to the front row ?
Hybrid translation	is it okay to the front row ?
Reference	can i go up to the front ?

Figure 5: *Comparison of reference, PBT and Hybrid translation outputs for several sentences.*

8. Acknowledgements

This work was partially funded by the European Comission (FEDER) and the Spanish Ministry of Education and Science (MEC) under the MIPRCV "Consolider Ingenio 2010" research programme (CSD2007-00018), the PROMETEO/2009/014 project and the iTransDoc project under grant TIN2006-15694-CO2-01; and by the Calencian "Conselleria d'Educació" under grant BFPI/2007/117.

9. References

[1] C. Callison-Burch, C. Fordyce, P. Koehn, C. Monz, and J. Schroeder, "Meta-evaluation of machine translation," in *Proceedings of the Third Workshop on Statistical Machine Translation*. Columbus, Ohio: Association for Computational Linguistics, June 2007, pp. 70–106.

[2] S. DeNeefe, K. Knight, W. Wang, and D. Marcu, "What can syntax-based mt learn from phrase-based mt?" *Proceedings EMNLP-CoNLL*, 2007.

[3] D. Chiang, "A hierarchical phrase-based model for statistical machine translation," in *ACL*, 2005.

[4] A. Zollmann and A. Venugopal, "Syntax augmented machine translation via chart parsing," in *Proceedings on the Workshop on Statistical Machine Translation*. New York City: Association for Computational Linguistics, June 2006, pp. 138–141.

[5] D. Wu, "Stochastic inversion transduction grammars and bilingual parsing of parallel corpora," *Computational Linguistics*, vol. 23, no. 3, pp. 377–403, 1997.

[6] C. Cherry and D. Lin, "A comparison of syntactically motivated word alignment spaces," *EACL*, pp. 145–152, 2006.

[7] H. Zhang, C. Quirk, R. C. Moore, and D. Gildea, "Bayesian learning of non-compositional phrases with synchronous parsing," in *Proceedings of ACL-08: HLT*. Columbus, Ohio: Association for Computational Linguistics, June 2008, pp. 97–105.

[8] F. J. Och and H. Ney, "Discriminative training and maximum entropy models for statistical machine translation," in *ACL*, 2002, pp. 295–302.

[9] P. Koehn, F. J. Och, and D. Marcu, "Statistical phrase-based translation," in *HLT/NAACL*, 2003.

[10] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, "Moses: Open source toolkit for statistical machine translation," in *Annual Meeting of the Assoc. for Computational Linguistics*, Prague, Czech Republic, 2007, pp. 177–180.

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

[12] O. F. Zaidan, "Z-MERT: A fully configurable open source tool for minimum error rate training of machine translation systems," *The Prague Bulletin of Mathematical Linguistics*, vol. 91, pp. 79–88, 2009.

[13] R. Levy and C. Manning, "Is it harder to parse chinese, or the chinese treebank?" in *ACL '03: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*. Morristown, NJ, USA: Association for Computational Linguistics, 2003, pp. 439–446.

[14] A. Stolcke, "Srilm – an extensible language modeling toolkit," in *International Conference on Spoken Language Processing.*, 2002.

[15] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. Morristown, NJ, USA: Association for Computational Linguistics, 2001, pp. 311–318.