ITI-UPV system description for IWSLT 2010

Guillem Gascó, Vicent Alabau, Jesús Andrés–Ferrer, Jesús González–Rubio, Martha–Alicia Rocha
Germán Sanchis–Trilles, Francisco Casacuberta, Jorge González, Joan–Andreu Sánchez

Instituto Tecnológico de Informática
Departamento de Sistemas Informáticos y Computación
Universidad Politécnica de Valencia
{ggasco, valabau, jandres, jegonzalez, mrocha}@dsic.upv.es
{gsanchis, fcn, jgonzalez, jandreu}@dsic.upv.es

Abstract

This paper presents the submissions of the PRHLT group for the evaluation campaign of the International Workshop on Spoken Language Translation. We focus on the development of reliable translation systems between syntactically different languages (DIALOG task) and on the efficient training of SMT models in resource-rich scenarios (TALK task).

1. Introduction

For this year's evaluation campaign of the International Workshop on Spoken Language Translation (IWSLT), the Pattern Recognition and Human Language Technologies (PRHLT) research group of the Universidad Politécnica de Valencia submitted runs for the Chinese–English spoken dialogs (DIALOG) task and English–French public speeches (TALK) task. In this paper, we report the configuration of such systems, together with preliminary experiments performed to establish the final setups.

Concerning the Chinese–English DIALOG task, we focus on the combination of different SMT systems with the purpose of combining the high coverage provided by phrase-based systems and the flexibility provided by syntax-based systems. We choose the median string algorithm to combine the different SMT systems into a single consensus translation [1, 2]. Our submission to the DIALOG task is the result of combining syntax-based and phrase-based models. Additionally, for the ASR output condition, an extra phrase-based model trained on ASR output lattices was added.

Regarding the English–French TALK task, our objective is to make good use of all the bilingual data provided without making use of too large amounts of computational resources, which may be unnecessary. A phrase-based model is trained using the TED corpus and some data from the additional training corpora. This additional data is selected to maximize translation quality and coverage of the phrase-based model. Additionally, Bayesian adaptation of model parameters is performed in order to provide stability to the results obtained. Such stability problems are often present whenever the size of the development data set is not large enough.

The *statistical machine translation* (SMT) problem can be stated as follows: given a sentence $\mathbf{f}$ from a certain source language, an equivalent sentence $\mathbf{e}$ in a given target language that maximizes the posterior probability is to be found. According to the Bayes decision rule, such statement can be specified as follows:

$$\hat{\mathbf{e}} = \underset{\mathbf{e}}{\operatorname{argmax}}\, Pr(\mathbf{e}|\mathbf{f}) \ . \qquad (1)$$

A direct modeling of the posterior probability $Pr(\mathbf{f}|\mathbf{e})$ has been widely adopted, and, to this purpose, different authors [3, 4] proposed the use of log-linear models, where

$$p(\mathbf{e}|\mathbf{f}) = \frac{exp \sum_{k=1}^{K} \lambda_k h_k(\mathbf{f}, \mathbf{e})}{\sum_{\mathbf{e}'} exp \sum_{k=1}^{K} \lambda_k h_k(\mathbf{f}, \mathbf{e}')} \ , \qquad (2)$$

and the decision rule is given by:

$$\hat{\mathbf{e}} = \underset{\mathbf{e}}{\operatorname{argmax}} \sum_{k=1}^{K} \lambda_k h_k(\mathbf{f}, \mathbf{e}) \ , \qquad (3)$$

where $h_k(\mathbf{f}, \mathbf{e})$ is a score function representing a feature for the translation of $\mathbf{f}$ into $\mathbf{e}$, K is the number of features and $\Lambda = [\lambda_1, \ldots, \lambda_K]^T$ are the weights for the log-linear combination.

The rest of the paper is organized as follows. Section 2 describes our approach for the Chinese–English DIALOG task. Section 3 presents our submission to the English–French TALK task. In Section 4, the systems employed in the evaluation campaign are sketched and results are presented and discussed. A summary and a list of related issues we will investigate in the next future end the paper.

2. Stochastic inversion transduction grammars for syntactically different languages

As in 2009, the central focus of the DIALOG task is the translation of task-oriented human dialogs in travel situations, between Chinese and English. The DIALOG task is carried out using the Spoken Language Databases (SLDB) corpus, a collection of human-mediated cross-lingual dialogs in travel situations. In addition, parts of the BTEC corpus are also provided to the participants of the DIALOG Task.

In this section we describe the ITI-UPV machine translation system used in the DIALOG Task of the IWSLT 2010. Syntax-based SMT has been successfully used for translating syntactically different language pairs such as English-Chinese [5]. For that reason, we used a syntax based decoder in this task. However, we did not want to loose the high coverage of the phrase-based systems so we combined the outputs of the syntax-based decoder with the outputs of a state of the art phrase-based system. Our submission is then, the result of combining the outputs of a syntax-based SMT system and a phrase-based SMT system. Additionally, for the ASR output condition, we add a third phrase-based SMT system trained on the ASR output lattices. The system combination approach is based on median string techniques [1].

2.1. ITG-based decoding in SMT

Inversion Transduction Grammars (ITGs) [6] are a restricted set of Synchronous grammars. Standard ITGs use only word-to-word transduction, however, in order to use the advantages of phrasal translation the original formalism has been extended to allow direct phrasal transductions.

An ITG with phrasal productions is a tuple $(N, \Sigma, \Delta, S, \mathcal{R})$ where N is the set of non-terminals, $S \in N$ is the root non-terminal, Σ is the source language alphabet, Δ is the target language alphabet, and $\mathcal{R}$ is a set of rules. Rules can be divided in two sets: syntactic rules and lexical rules. Syntactic Rules have the form: $A \to [BC]$ or $A \to \langle BC \rangle$, where A, B and C are non-terminals and the brackets enclosing the right part of the rule (direct rules) mean that the two non-terminals are expanded in the same order in the input and output languages, whereas the rules with pointed bracketing (inverse rules) expand the left symbol into the right symbols in the straight order in the input language and in reverse order in the output language. Lexical Rules have the form $A \to x/y$ where $x \in \Sigma^*$ and $y \in \Delta^*$. Σ^* and Δ^* are the free monoids[1]. It must be noted that x or y can be the empty string, denoted by ϵ, which is not allowed in both sides of the same production.

Stochastic ITGs are the probabilistic extension of ITGs, in which each rule has a probability attached. A derivation is a sequence of rules that, from the initial non-terminal, expand to one string of source language terminals and one of target language terminals. The probability of a derivation is the product of the probabilities of each of its rules.

The SITG formalism can be used as a translation model: given one source language sentence $\mathbf{f}$ the system must find a target language sentence $\hat{\mathbf{e}}$ that maximizes the probability of a complete derivation that yields the bilingual sentence $(\mathbf{f}, \hat{\mathbf{e}})$. We can obtain also the resulting derivation $\hat{D}$. That is

$$(\hat{\mathbf{e}}, \hat{D}) = \underset{(\mathbf{f}, D)}{\operatorname{argmax}} \Pr(S \Rightarrow^D (\mathbf{f}, \mathbf{e})) \,. \quad (4)$$

In order to increase the performance of the decoder we added several additional models commonly used in SMT (n-gram language model, lexical models...) and we combine them using a log-linear combination of probability models.

During the decoding process, the source language sentence is split in phrases that are translated using the lexical rules of the SITG and then merged in a straight or inverted order using the syntactic rules. The search algorithm used in the decoder is similar to the CYK parsing algorithm for context-free grammars [7] but storing in each cell of the chart, not only the non-terminals, but also the partial translation hypotheses. The use of n-gram language models has been demonstrated to be very useful for phrase-based 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 its use is not enough to get the most likely translation. In order to partially alleviate this problem, we need to use a translation hypotheses stack in each cell of the CYK-like chart instead of a single hypothesis. The hypotheses of two stacks can be combined directly or inversely, and the n-gram language model score of the new resulting hypotheses must be recomputed.

The system implements two different kinds of pruning:

1. Histogram Pruning: In each agenda only the n most likely hypotheses are stored.

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

Both pruning strategies are parameterizable, so it can be chosen between a slow but precise search or a fast and more inaccurate one.

In order to obtain an ITG with linguistic information from the bilingual corpus provided, we used the method explained in [8].

2.2. Lattices for ASR error recovery

A lattice $\mathcal{L}$ is a compact representation of the (pruned) search space of the speech recognizer. A lattice stores multiple hypotheses of the ASR system and provides a convenient representation for tight ASR and SMT coupling. Although there are several studies dealing with algorithms for translating ASR lattices [9, 10], their practical use is limited because of the computational resources they need.

[1] The set of all finite-length strings on Σ and Δ respectively.

Conversely, Confusion Networks (CNs) are even a more compact representation of the hypothesis space for which efficient SMT decoding algorithms exist [11]. CNs attempt to minimize the expected number of word errors, computed as the number of substitutions, deletions and insertions needed to transform the hypothesis into the reference. They can be obtained from lattices by aligning words from the different lattice hypotheses into a flat sequential lattice. The arcs of the CNs between the node i and the node $i+1$ represent competing words at position i with the word posterior probability $Pr(f|i, \mathcal{L})$. In the decoding process, these word posterior probabilities are combined with the translation probabilities in such a way that translation performance is optimized.

2.3. Median string computation for system combination

The different SMT models trained are combined into a consensus translation that takes advantage of the strengths of the individual systems and smooths their limitations. The consensus translation is computed as the median string over the translations of the individual systems [1, 2].

Given a set $E = \{\mathbf{e}_1 \ldots \mathbf{e}_n \ldots \mathbf{e}_N\}$ of translations from N MT systems, let Δ be the vocabulary in the target language ($E \subseteq \Delta^*$). The median string of set E is given by:

$$\mathcal{M}(E) = \underset{\mathbf{e}' \in \Delta^*}{\text{argmin}} \sum_{n=1}^{N} \mathcal{D}(\mathbf{e}', \mathbf{e}_n) \ , \qquad (5)$$

where $\mathcal{D}(\cdot, \cdot)$ is a string distance function. We choose to use the normalized edit distance [12] in our submission. The normalized edit distance had been successfully applied in the computation of median strings in different classification and system combination tasks [13, 1, 2].

Computing the median string is a NP-hard problem [14], therefore, only approximations to the median string can be computed in reasonable time. In our submission, the median string is computed by means of the *approximate median string* algorithm [13]. The approximate median string algorithm starts with an initial string that is iteratively improved by successive refinements. This refinement process is based on the greedy application of edit operations[2] over this initial string looking for a reduction of the accumulated distance to the translations in the set.

The initial string of the algorithm can be a random string, one of the translations in the set or even an empty sting. Starting with a better initial string results in fewer iterations for the algorithm to converge, but the different initializations do not affect the performance of the median string computed [13]. We took, from the given translations, the one with the lowest accumulate distance as the initial string of the algorithm. Then, the procedure described in Algorithm 1 is repeated until there is no improvement.

```
For each position $i$ in the string $\mathbf{e}$:
  1. Build alternatives:
       Substitution: Make $\mathbf{x} = \mathbf{e}$. For each word $w \in \Delta$:
         • Make $\mathbf{x}'$ the result string of replacing $x_i$ by $w$.
         • If the accumulated distance of $\mathbf{x}'$ to $E$ is lower than
           the accumulated distance of $\mathbf{x}$, then make $\mathbf{x} = \mathbf{x}'$.
       Deletion: Make $\mathbf{y}$ the result string of deleting $e_i$ from $\mathbf{e}$.
       Insertion: Make $\mathbf{z} = \mathbf{e}$. For each word $w \in \Delta$:
         • Make $\mathbf{z}'$ the result string of inserting $w$ at place $i$ on $\mathbf{z}$.
         • If the accumulated distance of $\mathbf{z}'$ to $E$ is lower than the
           accumulated distance of $\mathbf{z}$, then make $\mathbf{z} = \mathbf{z}'$.
  2. Choose an alternative:
         • From the set $\{\mathbf{e}, \mathbf{x}, \mathbf{y}, \mathbf{z}\}$ take the string $\mathbf{e}'$ with less
           accumulated distance to $E$. Make $\mathbf{e} = \mathbf{e}'$.
```

Algorithm 1: Iterative process to obtain the approximate median string. Different edit operations are applied over each position of the string $\mathbf{e}$. The edited string with the lower accumulated distance to the set E is returned. The process is repeated until there is no improvement.

3. Sentence selection in resource-rich scenarios

In this section, we describe the ITI-UPV machine translation system that has been designed for the IWSLT' 10 TALK task. A state-of-the-art phrase-based SMT approach has been followed in order to translate English text subtitles into French. Specifically, our translation engine is the Moses system [15].

Within this task, the TED corpus is the in-domain data and is composed of a collection of English-French subtitles. However, the additional corpora are composed of sentences. In order to take advantage of the available extra training data, a homogeneous translation framework has then to be defined. We decided that our system would be based on sentences (rather than on subtitles) and so TED data were processed in that sense. Subtitle recovery is then needed as a post-process to SMT. All the steps are detailed in the next subsection.

3.1. Subtitle segmentation recovery

The TED corpus is a corpus segmented at subtitle level, not sentence level. Several subtitles can form a meaningful sentence. When dividing a sentence into several subtitles, we can loose some valuable context information. This has motivated a strategy to concatenate several bilingual subtitles into a sentence and finally, after the translation, to recover the original subtitle segmentation.

However, we do not want to allow reorderings between different subtitles within a single sentence. For that reason, we use the Moses XML tag <wall />. Our strategy for this translation task consists of the application of a sequential process that is composed of the following steps:

1. Sentence composition. Subtitles are concatenated to compose meaningful sentences so that context be-

[2] Insertion, deletion and substitution of single words.

tween consecutive subtitles is not lost. To that end, several linguistic rules have been adopted, which establish how subtitles may be grouped into sentences. Several subtitles are concatenated into a sentence until one of them ends with an "end of line" punctuation mark. The "end of line" punctuation marks taken into account are ".", "?" and "!". In addition, in order to consider a subtitle as the end of a sentence, the punctuation mark of the end must be present in both languages (English and French).

2. Segmentation and Translation. Although subtitles were merged in order to build independent sentences, the information about their subtitle composition is kept by means of a <wall /> XML tag. This label is used to mark the union point between subtitles in a sentence. Moses is able to process that input in such a way that subtitles are translated as a block.

3. Subtitle recovery. If Moses is appropriately employed, it reports the segmentation or alignment at phrase level that relates both the source sentence and its translation. Since the <wall /> tag forbids the reorderings around it, the alignment information allows us to determine where the translation of every subtitle starts and ends. Thus, the original subtitle segmentation can be restored in the translated sentences.

3.2. Probabilistic sentence selection

In the TALK task, we had to face the problem of using several training corpora, some of which are out-of-domain. This scenario posed a very appealing problem, i.e., how several information sources from different domains and sizes, can be used in order to find a trade-off between resources and performance.

The proposed approach consists in trying to conserve the probability distribution of which the in-domain corpora is assumed to be a representative sample. For doing this, it is mandatory to exclude sentences from the out-of-domain corpora, specifically, those that would distort the in-domain probability the most. In other words, we developed a sentence selection framework in which the training set is split into two corpora:

- *In-domain corpora:* the part of the corpora that shares the domain with the test sentences, and
- *Out-of-domain corpora:* the part that belongs to other domains.

It is assumed that there are not enough resources available to process all the corpora; or that by doing so, the system performance may be decreased, due to differences between the in-domain and the out-of-domain corpora. The proposed approach consists in approximating the in-domain probability distribution and, then, sampling sentences from the out-of-domain corpora accordingly to the approximated *in-domain* probability distribution.

The in-domain probability was approximated as follows:

$$p(\mathbf{e}, \mathbf{f}, |\mathbf{e}|, |\mathbf{f}|) = p(\mathbf{e}, \mathbf{f}/|\mathbf{e}|, |\mathbf{f}|)p(|\mathbf{e}|, |\mathbf{f}|) \qquad (6)$$

where the length distribution was computed by maximum likelihood estimation applied to the in-domain training corpus; and the in-domain translation probability, $p(\mathbf{e}, \mathbf{f}/|\mathbf{e}|, |\mathbf{f}|)$, was approximated by a log-linear model:

$$p(\mathbf{e}, \mathbf{f}/|\mathbf{e}|, |\mathbf{f}|) = \frac{1}{\mathcal{Z}(\mathbf{e}, \mathbf{f})} \exp(\sum_k \lambda_k h_k(\mathbf{e}, \mathbf{f})) \qquad (7)$$

where $\mathcal{Z}(\mathbf{e}, \mathbf{f})$ stands for the normalization constant. As for the features $h_k(\ldots)$, we used: a direct and an inverse IBM model 4 [16]; and both, source and target, 5-gram language models. All previous feature models are estimated using the in-domain corpora. Although computing the optimum λ_k weights may have some interest, in this first approach all weights were set to 1.

3.3. On-line sentence selection for infrequent n-grams recovery

When a source language n-gram appears few times in the training corpus, its alignment with the corresponding target language cannot be computed accurately. The problem is even worse when the n-gram does not appear (in the case of 1grams, it is considered an out of vocabulary word). As we stated in the previous subsection, only a small part of all the possible training data is used. Thus, some important information for the test set translation can be found in the discarded sentences. In this subsection, we explain a method to recover the sentence pairs from the discarded corpus that contain infrequent n-grams important to translate the test set.

A sentence pair is considered important for the test translation when it contains n-grams of the source language test sentences that are infrequent (or even non existent) in the selected training set. An n-gram is infrequent when it appears in the training set less than t times. Thus, each n-gram x of the source sentences of the test set is scored using the following function:

$$\mathrm{s}(x) = \max\{0, t - N(x)\} \qquad (8)$$

where $N(x)$ is the number of times the n-gram x appears in the training corpus. All the n-grams that do not appear in the test set have an score of 0. The sentences e discarded for the training can be scored with:

$$\mathrm{sc}(e) = \sum_{0 \leq i < j \leq |e|} \mathrm{s}(e_i^j) \qquad (9)$$

where e_i^j is the n-gram corresponding to the source language sentence positions from i to j.

Once all the discarded sentences are scored, the sentences with the highest score are incorporated into the training set. In order to avoid the inclusion of a lot of sentences with the same n-grams, the score of the sentences is computed dynamically, so that the counts of an n-gram in the training ($N(x)$) are recomputed each time a new sentence is included in the training.

One last consideration is that including too many sentences selected with the strategy presented here may alter significantly the probability distribution underlying the training data available. This is, in general, not a good idea, and for this reason it is only prudent to use the strategy presented here to introduce into the translation system only a small amount of bilingual samples.

3.4. Bayesian adaptation for model stabilization

Log-linear weights are typically estimated by means of the MERT [4] algorithm. However, this approach often shows stability issues whenever the amount of development data is not big enough. For this reason, we analyzed the effect of applying Bayesian adaptation [17] for stabilizing the log-linear weights involved in the translation process. Under the Bayesian adaptation paradigm, model parameters (i.e. log-linear weights in this case) are viewed as random variables having some kind of a priori distribution. Following the derivation presented in [17], the decision rule in Section 1 is re-written as:

$$\hat{\mathbf{e}} = \operatorname*{argmax}_{\mathbf{e}} \Pr(\mathbf{e}|\mathbf{f};T,A) \tag{10}$$

where T is the training data and A the adaptation data. Then, $\Pr(\mathbf{e}|\mathbf{f};T,A)$ is computed as follows:

$$\begin{aligned} p(\mathbf{e}|\mathbf{f};T,A) &= \mathcal{Z}' \int p(A|\Lambda;T)p(\Lambda|T)p(\mathbf{e}|\mathbf{f},\Lambda) \\ &= \mathcal{Z}' \int \prod_{\forall a \in A} \frac{\exp \sum_k \lambda_k \, f_k(\mathbf{f}_a, \mathbf{e}_a)}{\sum_{\mathbf{e}'} \exp \sum_k \lambda_k \, f_k(\mathbf{f}_a, \mathbf{e}')} \\ &\quad \exp\left\{ -\frac{1}{2}(\Lambda - \Lambda_T)^T \sigma_T^{-1} (\Lambda - \Lambda_T) \right\} \\ &\quad \frac{\exp \sum_k \lambda_k \, f_k(\mathbf{f}, \mathbf{e})}{\sum_{\mathbf{e}'} \exp \sum_k \lambda_k \, f_k(\mathbf{f}, \mathbf{e}')} \, d\lambda \end{aligned} \tag{11}$$

where the first term in the integral stands for the probability of the adaptation data, the second term is the prior distribution of the model parameters, and the last term is the probability of the sentence which is currently being translated.

Since computing the integral over the complete parametric space is unfeasible, a random sampling of these parameters was performed by choosing alternatively only one of the weights in $\mathbf{\Lambda}_t$, and modifying it randomly within a given interval. This being done, $\mathbf{\Lambda}_t$ was renormalized accordingly.

Moreover, the sum $\sum_{\mathbf{e}'}$ is approximated as the sum over all the hypothesis within the n-best list generated by the decoder, and instead of performing a full search we will perform a re-rank of the n-best list according to Equation 11.

Since complete coverage of all sentence pairs is not guaranteed by state-of-the-art SMT systems, $\mathbf{e}_a$ in Equation 11 is replaced by $\mathbf{e}_a^*$, which is the best hypothesis the search algorithm is able to produce, according to a given translation quality measure.

Subset	Language	$\|S\|$	$\|W\|$	$\|V\|$
train	English	30K	330K	7928
	Chinese	30K	272K	9891
zh-en dev	English	200*	2457.75*	436*
	Chinese	200	2140	379
en-zh dev	English	210	3095	399
	Chinese	210*	2661.5*	624*

Table 1: Main figures of the DIALOG corpus. The numbers with * are computed over multi-reference sets (average for sentences and running words, and total size of vocabulary).

Once normalization terms have been removed, and the above-mentioned approximations have been introduced, $p(\mathbf{e}|\mathbf{f};T,A)$ is no longer a probability. Hence, a leveraging term δ is introduced, and the final formula for $p(\mathbf{e}|\mathbf{f};T,A)$ is

$$p(\mathbf{e}|\mathbf{f};T,A) = \sum_{\Lambda_m \in MC(\Lambda_t)} (p(A|\Lambda;T)p(\mathbf{e}|\mathbf{f},\Lambda))^{\frac{1}{\delta}} \, p(\Lambda|T) \tag{12}$$

where $MC(\Lambda_T)$ is the set of Λ_m weights generated by the above-mentioned random procedure.

4. Evaluation results

In all the tables reported in this section, K stands for thousands of elements, $|S|$ stands for the number of sentences within a corpus, $|W|$ is the number of running words, and $|V|$ the vocabulary size. In addition, corpus statistics are reported on the tokenized and lowercased corpora, and after filtering sentences considered too long (i.e. more than 40 words).

4.1. Baseline system

For building the initial SMT systems, the open-source SMT toolkit Moses [15] was used in its standard setup. The decoder features a log-linear model comprising a phrase-based translation model, a language model and a lexicalized distortion model. The translation model, in turn, comprises direct and inverse phrase-translation probabilities, lexicalized weights, and word and phrase penalties. Phrases were obtained from symmetrized word alignments generated by means of GIZA++ [18]. In the baseline setup, the weights of the log-linear interpolation were optimized by means of ME RT [4]. In addition, a 5-gram LM with Kneser-Ney [19] smoothing and interpolation was built by means of the SRILM [20] toolkit.

4.2. DIALOG: Chinese–English system

Table 1 shows the main statistics for the official training train and development dev partitions. Note that each of the translation directions has a different development set.

In order to cope with this task, we used a combination of several systems.

Subset	Language	$\|S\|$	$\|W\|$	$\|V\|$
`train`	English	47.5K	747.2K	24.6K
	French	47.5K	792.9K	31.7K
`indev`	English	571	9.2K	1.9K
	French	571	10.3K	2.2K
`ofdev`	English	641	12.6K	2.4K
	French	641	12.8K	2.7K

Table 2: Main figures of the TALK corpus.

The first system is the phrase-based system Moses [15]. In order to train this system, we included all the development sets into the training data. Moreover, since the development sets are multi-reference, we decided to include all the references in the training for the phrase-based models but not for the language model since duplicating sentences distorts the LM estimates. The phrase-based models are not significantly distorted by this duplications and they are meaningfully enriched by synonym phrases.

Based on previous experimentation [8], we decided to use linguistic information in the ITG-based system. In order to train the ITG, we used the Chinese and English versions of the Stanford Parser [21]. All the development sets were included in the training of the final system.

The system for ASR and SMT coupling was created following the next procedure. To begin with, a phrase-based Moses system was trained for which the source was preprocessed to form sentences resembling the ASR input. To do that, the source was tokenized, then converted to lowercase and, finally, all punctuation marks were removed. In addition, several words were substituted to a normalised form since there was a mismatch between their training and ASR representations. Secondly, the lattices were preprocessed as well. After the words in the lattices were tokenized, a CN was created from them using the SRILM toolkit [20]. At last, we used the CN translation decoder [11] implemented in Moses to perform the ASR and SMT integration.

Given that we have only two systems to be combined (three for the ASR output condition), we choose to combine the 20-best translations of each individual system rather than using only the single best ones. Therefore, 40 (60 for the ASR output condition) hypotheses are combined to obtain our final submission. All the combinations were computed by means of the approximate median string algorithm.

4.3. TALK: English–French system

For internal development purposes and because the final test set was not released until about one month before the final translations were due, we decided to split the training set provided for the task in two different subsets: one for training and a smaller one for internal development purposes. We will name the training set `train`, the internal development set `indev`, whereas the official development set, which was used as test set until the final test set was released, will be named `ofdev`. Statistics can be seen in Table 2.

Corpus	Language	$\|S\|$	$\|W\|$	$\|V\|$
Europarl	English	1.25M	25.6M	81.0K
	French	1.25M	28.2M	101.3K
News Commentary	English	67.6K	1.4M	35.6K
	French	67.6K	1.6M	43.3K
United Nations	English	5.0M	94.4M	302.7K
	French	5.0M	107.4M	283.7K
Gigaword	English	15.5M	302.9M	1.6M
	French	15.5M	360.6M	1.6M

Table 3: Main figures of the out-of-domain corpora provided for the English→French TALK task.

In addition to the TALK corpus, larger out-of-domain corpora were also provided. Statistics of such corpora are provided in Table 3. As it can be seen, these corpora are farely big, and the amount of data available is enough to overwhelm the amount of data of the in-domain corpus. For this reason, and as described in Section 3.2, we confronted the problem as a data selection task. For this purpose, we considered all four out-of-domain corpora as a single, very large corpus, from which appropriate sentences were selected according to the strategies described in Sections 3.2 and 3.3.

The results of applying the probabilistic sentence selection technique described in 3.2 can be seen in Table 4. For each one of the systems presented in this table, MERT was re-run in order to optimize the log-linear weights according to the `indev` set.

nK	BLEU	TER
0	23.2	60.8
10	23.5	60.5
50	24.2	59.8
100	25.0	59.0
200	25.1	58.9
500	25.5	58.7

Table 4: Evolution of translation quality according to the amount nK (in thousands) of sentences added to the baseline system. Results are given in terms of BLEU and TER evaluated on the `ofdev` set.

Results of applying the on-line selection strategy described in Section 3.3 can be seen in Table 5. Specifically, and in order to avoid introducing too much distortion into the system, we decided to introduce with this technique a maximum of 10% of the amount of sentences introduced with the probabilistic technique. Several things should be noted:

- Translation quality when optimizing the log-linear weights by means of MERT shows a fairly unstable behavior, and no real conclusion about the effect of the on-line sentence selection technique can be drawn.

- When using a constant set of weights, estimated on the development set of the Europarl data and with a translation model trained only on Europarl data, the technique described seems to be beneficial, achieving improvements of up to 0.6 BLEU points when adding a total of 10K sentences (for $nK = 100$ and $t = 25$). However, in some cases the translation quality obtained is slightly lower than in the case of MERT, e.g. in the case of $nK = 50$ and $t = -$.

- When applying Bayesian adaptation for stabilizing the log-linear weights, we obtain the best performance in any case, both when comparing with MERT or `Euro`. Improvements prove to be consistent when increasing t, which seems reasonable. In this case, the Gaussian for the prior of the Bayesian adaptation technique was centered on the weights estimated for Europarl, since they can be considered to be well estimated on a 2000 sentence development set. δ was set to 32 and the size of the n-best list considered was 200, both according to the experiments reported in [17].
- Although in some cases the total number of sentences in the system is the same (e.g. for $nK = 50$; $t = 10$ and $t = 25$), it should be emphasized that the specific sentences included are not the same, since sentences are added until the maximum amount allowed is reached.
- Experiments with other nK values were not performed because of time constraints.

At the sight of these results, the final system submitted for final evaluation was built by sampling a total of 500K sentences from the out-of-domain corpora, and then adding 50K sentences by means of the on-line sentence selection technique, with $t = 10$. Both `indev` and `ofdev` were also included into the system, adding up to a final total of 645.6K sentences. The test set was translated using the log-linear weights estimated for the Europarl development data and Bayesian adaptation was applied by considering the `ofdev` set as adaptation data. For translating the `ofdev` set for the purpose of selecting $\mathbf{e}^*$ as described in Section 3.4, the `ofdev` set itself was not included into the system in order to avoid over-training.

5. Conclusions

With respect to the DIALOG task, the consensus translations computed take advantage of the high coverage of phrase-based models and the good performance on syntactically different languages of SITGs. In addition, for the ASR output condition, the use of the CN translation system helped the 1-best systems to recover from ASR errors.

In the TALK task, it has been shown that an intelligent selection of training data might be a good strategy towards a better utilization of computational resources. Specifically, the final system presented by the PRHLT group ranked only two BLEU points below the best system, by only using about 3% of the available training sentences. Of course, it is still necessary to know whether such best system also applied some kind of data selection strategy. In addition, it has also been shown that Bayesian adaptation can be applied in order to stabilize the log-linear weights within the core of a SMT system, and that such stabilization can be very helpful towards elucidating whether a given technique provides improvements or not.

nK	t	$\|S\|$	MERT	Euro	bayes
50	-	96.9K	24.2/59.8	24.5/59.8	24.7/58.7
	1	99.9K	23.7/60.6	24.5/59.6	24.9/58.8
	10	101.9K	24.1/60.5	24.8/59.4	25.2/58.4
	25	101.9K	24.1/60.3	24.7/59.3	25.2/58.4
100	-	146.9K	25.0/59.0	24.6/59.0	25.1/58.6
	1	149.8K	24.6/59.6	24.9/59.3	25.3/58.5
	10	156.9K	24.1/60.2	25.0/59.3	25.4/58.3
	25	156.9K	24.6/59.4	25.2/59.2	25.5/58.4

Table 5: Evolution of translation quality, as measured by BLEU/TER, when considering the on-line sentence selection technique described. `nK` stands for the amount of sentences added to the system with the probabilistic technique described in Section 3.2. `t` is the threshold described in Section 3.3. `-` means that the on-line technique was not applied at all. In the `MERT` column, translation quality when optimizing the log-linear weights by means of MERT over the `indev` set is shown, whereas `Euro` is in the case of considering the weights optimized for the Europarl corpus and `bayes` displays the results of applying the Bayesian adaptation technique described in Section 3.4.

6. Future work

In the future, we plan to study the use of lexicalized maximum entropy models for the reordering of hypotheses in the ITG-based decoder. In addition, the syntactic information used in the process of decoding can be used to find and correct grammatical errors in the output sentences.

Concerning the integration of the ASR and SMT systems, we would like to compare translation of CNs with different techniques of translating lattices. Although the complexity of the latter is higher, improvements in the output quality are expected. We are specially interested in using lattice confidence measures, which take into account multiple sources of information regarding the input.

Regarding the system combination technique, we are interested in modifying the approximate median string algorithm by allowing, in addition to substitutions, deletions and insertions of single words, shifts of word sequences. The use of other different distance functions will also be studied.

With respect to the sentence selection strategies described in this paper, there is still a lot of work to be done. Specifically, we would like to assess whether such selection strategy proves to outperform a random selection strategy. Although preliminary results seem to point in that direction, more experimentation is still needed in order to confirm such conclusion. Other future work involves investigating the effect of optimizing the log-linear combination applied for approximating the probability of a given sentence pair. In addition, more work is also needed towards establishing a good proportion between the amount of sentences added by means of the probabilistic sentence selection and the infrequent n-gram recovery techniques. With respect to Bayesian adaptation as applied for model stabilization, the results shown in this paper seem very promising, moreover considering that the different parameters present in the Bayesian adaptation framework were chosen based on very different work.

7. Acknowledgments

This paper is based upon work supported by the EC (FEDER/FSE) and the Spanish MICINN under the MIPRCV "Consolider Ingenio 2010" program (CSD2007-00018),iTrans2 (TIN2009-14511) project, and the FPU scholarship AP2006-00691. This work was also supported by the Spanish MITyC under the erudito.com (TSI-020110-2009-439) project and by the Generalitat Valenciana under grant Prometeo/2009/014 and scholarships BFPI/2007/117 and ACIF/2010/226 and by the Mexican government under the PROMEP-DGEST program.

8. References

[1] J. González-Rubio and F. Casacuberta, "On the use of median string for multi-source translation," in *Proc. of ICPR*, August 23-26 2010, pp. 4328–4331.

[2] J. González-Rubio, J. Andrés-Ferrer, G. Sanchis-Trilles, G. Gascó, P. Martínez-Gómez, M. Rocha, J. Sánchez, and F. Casacuberta, "UPV-PRHLT combination system for WMT 2010," in *Proc. of the Workshop on Statistical Machine Translation and Metrics (ACL)*, July 15–16 2010, pp. 296–300.

[3] K. Papineni, S. Roukos, and T. Ward, "Maximum likelihood and discriminative training of direct translation models," in *Proc- of ICASSP*, 1998, pp. 189–192.

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

[5] D. Marcu, W. Wang, A. Echihabi, and K. Knight, "SPMT: Statistical machine translation with syntactified target language phrases," in *Proc. of EMNLP*, 2006, pp. 44–52.

[6] D. Wu, "Stochastic inversion transduction grammars and bilingual parsing of parallel corpora," *Computational Linguistics*, vol. 23, no. 3, pp. 377–404, 1997.

[7] T. Kasami, "An efficient recognition and syntax-analysis algorithm for context-free languages," in *Scientific report AFCRL-65-758, Air Force Cambridge Research Lab, Bedford, MA.*, 1965.

[8] G. Gascó and J. A. Sánchez, "Syntax augmented inversion transduction grammars for machine translation," in *Proc. of CICLING*, 2010, pp. 427–437.

[9] V. Alabau, A. Sanchis, and F. Casacuberta, "Improving speech-to-speech translation using word posterior probabilities," in *Proc. of MT SUMMIT*, Copenhagen, Denmark, September 10-14 2007.

[10] C. Dyer, S. Muresan, and P. Resnik, "Generalizing word lattice translation," in *Proc. of ACL*, Columbus, Ohio, June 2008, pp. 1012–1020.

[11] N. Bertoldi and M. Federico, "A new decoder for spoken language translation based on confusion networks," 2005, pp. 86–91.

[12] E. Vidal, A. Marzal, and P. Aibar, "Fast computation of normalized edit distances," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 9, pp. 899–902, 1995.

[13] C. Martínez, A. Juan, and F. Casacuberta, "Use of median string for classification," in *Proc. of ICPR*, vol. 2, 2000, pp. 907–910.

[14] C. de la Higuera and F. Casacuberta, "Topology of strings: Median string is NP-complete." *Theoretical Computer Science*, vol. 230, pp. 39–48, 2000.

[15] 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 *Proc. of ACL*, 2007, pp. 177–180.

[16] P. F. Brown, S. D. Pietra, V. J. D. Pietra, and R. L. Mercer, "The mathematics of statistical machine translation: Parameter estimation," *Computational Linguistics*, vol. 19, no. 2, pp. 263–311, 1994.

[17] G. Sanchis-Trilles and F. Casacuberta, "Log-linear weight optimisation via bayesian adaptation in statistical machine translation," in *Proc. of COLING*, Beijing, China, August 23–27 2010.

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

[19] R. Kneser and H. Ney, "Improved backing-off for m-gram language modeling." *Proc. of ICASSP*, vol. II, pp. 181–184, May 1995.

[20] A. Stolcke, "SRILM – an extensible language modeling toolkit," in *Proc. of ICSLP*, 2002.

[21] R. Levy and C. Manning, "Is it harder to parse chinese, or the chinese treebank?" in *Proc. of ACL*, 2003, pp. 439–446.