TALP: Xgram-based Spoken Language Translation System

Adrià de Gispert and José B. Mariño

TALP Research Center
Universitat Politècnica de Catalunya, Barcelona
{agispert|canton}@gps.tsc.upc.es

Abstract

This paper introduces TALP, a speech-to-speech statistical machine translation system developed at the TALP Research Center (Barcelona, Spain). TALP generates translations by searching for the best scoring path through a Finite-State Transducers (FSTs), which models an Xgram of the bilingual language defined by tuples. A detailed description of the system and the core processes to train it from a parallel corpus are presented. Results on the Chinese-English supplied task of the Int. Workshop on Spoken Language Translation (IWSLT'04) Evaluation Campaign are shown and discussed.

1. Overview of the system

TALP (Traducció Automàtica del Llenguatge Parlat) is a speech-to-speech statistical machine translation system developed at the TALP Research Center (Barcelona, Spain) during the last years. It implements an integrated architecture by joining speech recognition and translation in one single step. Mathematically, the system produces a translation by maximizing the *joint* probability between source and target languages, which is equivalent to a language model of an special language with bilingual units (called tuples). TALP implements this tuple language model by means of a Finite-State Transducer (FST) considering an Xgram memory, that is, a variable-length N-gram model which adapts its length to evidence in the data. Xgrams have proved good results in speech recognition tasks in the past [1].

Given such a bilingual FST, the search for a translation becomes the search for the best-scoring path among the transducer's edges. This search can be performed by dynamic programming, using well-known decoding techniques from the speech recognition domain. This way, the Viterbi algorithm and a beam search can be used forwards taking only source-language words into account (first part of each tuple), reading words in the target language during trace-back to produce the translation. Using the same structure and search method, acoustic models can be omitted to perform text translation tasks only.

This work has been partially supported by the Spanish Government under grant TIC2002-04447-C02 (ALIADO project), the European Union under grant FP6-506738 (TC-STAR project) and the Dep. of Universities, Research and Information Society (Generalitat de Catalunya).

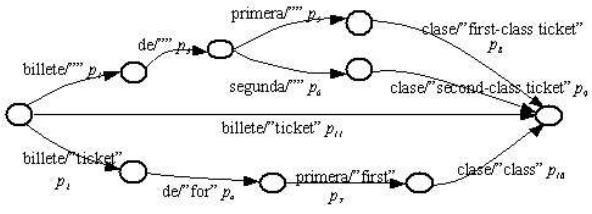

Figure 1: *A translation FST from Spanish to English*

This translation FST is learned automatically from a parallel corpus in three main steps (and an optional preprocessing). First, an automatic word alignment is produced. Currently this is done by the freely-available GIZA++ software [2], implementing well-known IBM and HMM translation models [3, 4]. From this alignment, a tuple extraction algorithm generates the set tuples that induces a sequential segmentation of both source and target sentences. These tuples must respect word order in both languages, as this is necessary for the transducer to produce a correct-order translated output. Finally, Xgrams are learned using standard language modeling techniques. Previous publications on this system include [5] and [6].

The organization of the paper is as follows. Section 2 offers an overview of the system architecture, whereas sections 2 and 3 deepen into details on translation generation and training issues. Section 4 presents the experimental framework used to evaluate the system, whose results are discussed in section 5. Finally, section 6 concludes and outlines future research lines.

2. Translation generation

Statistical machine translation is based on the assumption that every sentence e in the target language is a possible translation of a given sentence f in the source language. The main difference between two possible translations of a given sentence is a probability assigned to each, which is to be learned from a bilingual text corpus. This probability can be modeled by a joint probability model of

source and target languages. In this case, solving the translation problem is finding the sentence in the target language that maximises equation 1. This probability can be approximated by an Xgram of a joint or bilingual language model, learned from a set of tuples, as expressed in equation 2.

$$\hat{e} = \arg\max_{e}\{p(e,f)\} = \cdots = \quad (1)$$

$$\arg\max_{e}\{\prod_{n=1}^{N} p((e,f)_n|(e,f)_{n-1},...,(e,f)_{n-X+1})\} \quad (2)$$

where:

$$(e,f)_n = (e_{i_n}...e_{i_n+I_n}, t_{j_n}...t_{j_n+J_n})$$

TALP implements this Xgram language model by means of finite-state transducer whose edges are labelled with tuples (as shown in figure 1). That is, each edge has a label that relates one or more words in the source-language to zero, one or more words in the target language. This way, some edges may have just one word in the source language whereas others may have more, and both be valid as long as the first word is equal to the input. Bearing this in mind, all well-known ASR decoding techniques can be used to find the best-scoring translation of a given sentence, once the transducer is built.

In a speech-to-speech translation framework, input data is the speech signal, so the objective of translation becomes the search for the sentence e in the target language the following equation:

$$\hat{e} = \arg\max_{e}\{p(e,f) \cdot p(x|f)\} \quad (3)$$

where we introduce the acoustic model $p(x|f)$ in the optimisation (x being the input acoustic signal). Therefore, by following this transducer-based approach, the same training and search techniques can be used to tackle both text and speech translation tasks. The following section goes into the details on how the FST is learned from a parallel textual corpus.

The current architecture of the TALP translator performs the search for the best translation in a monotonous fashion. Any reordering of the target words is restricted to the short region defined by the tuple. That is, it can only be produced inside a tuple, which can contain crossed alignment relationships. This poses a strong limitation to the system, specially when dealing with pairs of languages with long reorderings in word alignment, such as Chinese and English. Several reordering techniques have been tested with the FST architecture, none of them providing significant results (for Spanish-English case, see [6]).

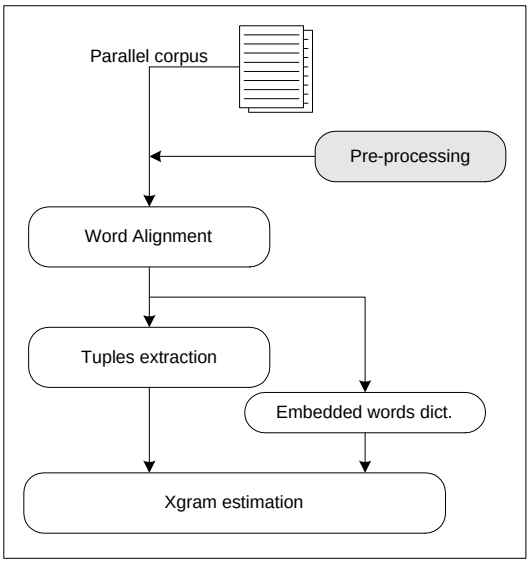

Figure 2: *Training stages from a parallel corpus to a translation FST*

3. Training

Usual language model techniques can be used to learn the tuples language model (X-gram) once a given parallel corpus has been transformed into a set of tuples for each sentence. In order to do so, the training of the system comprises three basic stages (and an optional preprocessing), which are shown in the flow diagram of figure 2. These steps are described in the following subsections.

3.1. Preprocessing

The pre-processing stage is aimed at categorising words in order to reduce output vocabulary, helping the alignment stage to increase accuracy, without reducing input flexibility. Some basic word groups can be categorised, namely personal names, names of cities, towns or countries (manually), and dates, times of the day and numbers (automatically). With the X-gram software, these categories or word groups can be easily modelled by smaller finite-state transducers that translate each of their possible alternative values.

However, this preprocessing is an optional and language-dependent stage, according to the availability of resources. In the frame of a Chinese-English translation task, only a small preprocessing has been performed. As evaluation is performed without punctuation marks, we experimented training without punctuation, but this was discarded as results were equal or worse than leaving punctuation until a final output post-processing.

On the other hand, a special segmentation of the training corpus was performed. Whenever a pair of Chinese-English sentences shared the same number and type of punctuation marks (considering '.' ',' and '?'), these were

split according to the position of punctuation. That causes the train corpus to have more and shorter sentences.

3.2. Word alignment

Assuming that the input parallel text in sentence aligned, we perform a standard statistical word alignment stage by using GIZA++, a freely-available software which implements the so-called IBM alignment models presented in [3] as well as the HMM-based alignment model [4], producing the Viterbi alignment as an approximation to the most probable one. Due to the asymmetric nature of the resulting alignment (linking one word in the source language to one or more words in the target language), several symmetrization strategies can be used (such as the union or the intersection between alignments in both directions).

In our case, both the union and the intersection are performed and can be also used to generate the set of tuples, like the source-to-target (s2t) and target-to-source (t2s) alignments.

3.3. Tuples extraction

Once the alignment is produced, the tuples extraction unit has to build units so that the order of the sentence in both languages is not violated, a necessary requirement when dealing with finite-state translation transducers, as already exposed in [7], because otherwise the transducer would learn order-incorrect sentences. Given a sentence pair and a corresponding word alignment, the sequential set of tuples contains those pairs of m source words and n target words satisfying these constraints:

1. It induces a monotonous segmentation of the pair of sentences.

2. Words are consecutive along both source and target sides of the tuple.

3. No word on either side of the tuple is aligned to a word out of the tuple.

4. Each tuple cannot be decomposed into smaller phrases without violating the previous constraint.

Note that this set is unique under these conditions [8]. The only ambiguity appears when a target word is aligned to NULL, in which case we append it to the next tuple (if exists, else to the previous). An example of the tuple extraction process is drawn in figure 3.

When extracting tuples with more than one word in each language (as the third tuple in figure 3), a certain local reordering of the target is necessarily encoded. While helping the system to avoid local reordering mistakes, this strategy can suffer from an information loss, as the source words appearing in this tuple may not have any

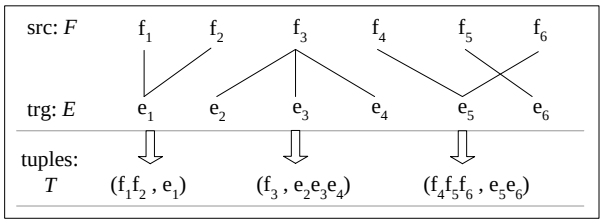

Figure 3: *Tuples extraction from an aligned sentence pair*

translation if they do not appear elsewhere *alone* in a tuple. We call these words *embedded*, as their translation appears only embedded in a longer phrase.

To avoid this, we build up a dictionary of translations for embedded words from the most accurate word alignment available. For a certain embedded word f_j and a given word alignment, we look for the target words $e_i...e_{i+K}$ that are most frequently aligned to f_j with these two conditions:

1. Target words $e_i...e_{i+K}$ are consecutive in the target sentence.

2. Target words are aligned *only* to f_j or to null.

This way, we build up a statistical dictionary independently of the non-monotonicity of the word alignment. The entries of the dictionary are used as unigrams in the bilingual model estimated by the FST. To create the dictionary, all four aforementioned word alignments have been tested for several translation tasks, and the intersection has consistently given better results, even though its translations are always one-word.

This strategy is useful though not robust enough yet. By building up the dictionary, we are able to produce a word-by-word translations for *some* embedded words whenever the sequence in the test sentence is not equal to any training tuple. However, information on embedded N-grams is not extracted at the moment. This has growing importance when dealing with very different pairs of languages, in terms of word ordering, as with a Chinese-English task. In section 4.3 the impact of this technique is evaluated in practice.

3.4. Xgram estimation

Finally, given the parallel corpus described in a set of tuples for each sentence, a Finite-State Transducer containing Xgram probabilities is learned. Usually, a maximum length of 3 is used for memory, to avoid over-fitting to training data. A back-off strategy is follow and a pruning of the resulting automaton can be performed. Two parameters are used for this: on the one hand, the minimum number of times a certain history (Xgram) must occur to be considered. And on the other hand, two different nodes sharing the same recent history are merged if the

divergence between their output probability distributions is smaller than a certain threshold (see details in [1]). Given the usual sparseness problems when dealing with parallel corpora, the first parameter is not used (set to 1), whereas the latter (hereafter referred to 'f') performs a slight pruning.

4. Experiment and results

The presented system has been evaluated in the framework of the International Workshop on Spoken Language Translation (IWSLT'04), a Satellite Workshop of the Interspeech - ICSLP. In the workshop, an Evaluation Campaign has been conducted for two translation directions, namely Chinese-to-English and Japanese-to-English. Moreover, two different tracks per direction have been proposed, namely using only the supplied corpus (supplied) and allowing the use of any additional data for training purposes (unrestricted). Besides, an intermediate track allowing the use of the supplied corpus plus certain linguistic resources available from LDC has been proposed for the Chinese-English task.

TALP has participated only in the Chinese-to-English supplied track, the reason being that we believe the Japanese-to-English task to be even more demanding in terms of reordering. As our system lacks any direct treatment of long reorderings, we found that the Chinese-to-English task brought up enough challenges for research. Next, we present a brief description of the supplied corpus, the evaluation measures used in the track and the results achieved by two different TALP runs.

4.1. Chinese-to-English IWSLT'04 supplied corpus

Table 1 shows the main statistics of the supplied data, namely number of sentences, words, vocabulary, and maximum and average sentence lengths for each language, respectively. The difference between 'Train set' and 'Segmented train set' is the segmentation discussed in section 3.1. A development set of 506 sentences was also supplied, together with 16 reference English translations. There are 160 unseen words in the development set and 104 unseen words in the test set.

4.2. Evaluation measures

The output of the system is evaluated using automatic and manual evaluation measures. For the automatic evaluation, 16 man-made English reference translations of the test corpus are used. The evaluation measures include BLEU score, NIST score, mWER, mPER and GTM (general text matcher).

As for human assessment, each translated sentence is evaluated by three human judges, according to "fluency" and "adequacy" of the translation. While fluency indicates how the evaluation segment sounds to a native speaker of English, from 'Incomprehensible' (1) to 'Flawless English' (5), adequacy judges how much of the information is carried by the translation, from 'None of it' (1) to 'All of the information' (5).

supplied	sent.	words	voc.	Lmax	Lavg
Train set					
Chinese	20,000	182,904	7,643	69	9.1
English		188,935	8,191	75	9.4
Segmented train set					
Chinese	22,205	—-	—-	62	8.2
English		—-	—-	58	8.5
Development set					
Chinese	506	3,515	870	24	6.9
Test set					
Chinese	500	3,794	893	62	7.5

Table 1: *Chi-Eng supplied corpus statistics*

4.3. Development work

Several different configurations were tested for the development set. Their results are shown in Table 2, where 'aU' and 'a2' refer to using the union and the s2t alignment, respectively. The term 'seg' refers to training with the segmented version of the corpus, whereas 'f' refers setting the pruning parameter to 0.2, instead of leaving it to 0 (see section 3.4). As about the effect of using a dictionary of embedded words (see section 3.3), an evaluation without it has been performed, leading to the results shown with term '-D'. In general, these results show a slight variation in performance for both alignments, but with a remarkable descent in terms of NIST score.

runs	BLEU	NIST	WER	PER	GTM
aU	0.244	5.169	0.615	0.529	0.591
aU,seg	0.251	5.187	0.607	0.521	**0.595**
aU,seg,f	0.255	**5.210**	**0.603**	**0.518**	0.594
aU,seg,-D	0.264	4.741	0.606	0.524	0.592
a2	**0.319**	3.789	0.614	0.552	0.573
a2,seg	0.318	3.871	0.606	0.546	0.573
a2,seg,f	0.314	3.678	0.607	0.548	0.570
a2,seg,-D	0.315	3.706	0.607	0.547	0.571

Table 2: *Automatic evaluation results (development set)*

All runs using the union alignment leave 7 sentences untranslated (empty), whereas runs using the s2t alignment leave 16, 18, 19 and 19 sentences each. As we can see, the greatest difference between all the results lies in the original word alignment used to extract the bilingual tuples. The segmented version of the corpus provides a slight but consistent improvement, helping the training to produce more accurate alignments and shorter tuples. As about pruning, it seems that this technique does not make

much of a change, but it turns the algorithm a bit more efficient.

4.4. Test set results

For the reasons presented above, we have selected configurations 'aU,seg,f' (run A) and 'a2,seg,f' (run B) as first-best and second-best for the test set. The difference in their original alignment makes a big difference in the final translation transducer, as we can see in the statistics shown in Table 3, where the total number of tuples, tuples vocabulary size, average tuples length (adding source and target words) and number of embedded words are shown.

runs	tuples	vcb	length	embed
A	97,338	27,039	3.9	4741
B	140,896	29,344	2.9	1545

Table 3: *Statistics of two different runs*

Usually, the union alignment leads to much longer tuples, which in turn increases the number of embedded words, whose translation is 'solved' by the dictionary built up with the intersection. On the contrary, using the s2t alignment we increase the number of total tuples (by decreasing their length), reducing at the same time the number of embedded words. However, we appreciate an important increase in the percentage of tuples translating to NULL (up to 28%, in contrast to 7.5% for the union), an undesirable consequence of following the asymmetric alignment. This could be avoided by taking a hard decision as to where to align these tuples (whether to the previous or the next tuple), but we do not believe this to be of much gain compared to using the union alignment. Finally, many of the new unigrams that are created through the dictionary when using the union alignment, already exist in the FST using the s2t alignment, but they are linked to NULL, which is inappropriate when no history can help in decoding. Table 4 presents the results obtained by these two runs evaluating against automatic measures.

runs	BLEU	NIST	WER	PER	GTM
A	0.279	6.778	0.556	0.465	0.647
B	0.331	5.391	0.550	0.490	0.620

Table 4: *Automatic evaluation results for two runs*

Run A has produced no output in 5/500 sentences. Run B has produced no output in 11/500 sentences. Results show a surprising behaviour: while NIST score, PER and GTM clearly prefer run A, the BLEU metric gives a much better score to run B, being the WER practically identical in both cases. All in all, we believe run A to be slightly better and more consistent with human translation, being more based on a phrase translation approach. It seems that BLEU does not seem to penalise the 'shortening' effect of run B output. In fact, the average output sentence length for run A is 6.01 words, whereas for run B is only 5.18, a clear consequence of the high percentage of tuples translating to NULL.

Table 5 presents the TALP results of the manual evaluation for run A. As expected given the lack of a reordering scheme in the statistical machine translator proposed, the fluency score does not even achieve a '3', meaning 'Non-native English'. However, the adequacy score is quite good as it means the 'Much of the information' is being translated in the output.

run	fluency	adequacy
A	2.792	3.022

Table 5: *Manual evaluation results for run A*

5. Analysis and discussion

Among the various configurations tested, the biggest difference lies in the word alignment used to extract tuples. However, all of them share a very important limitation of the current architecture. This refers to word reordering, which is strictly limited to the local reordering inside a tuple, making the approach inappropriate for pairs of languages with a very different word order. In the Chinese-English case, the system is unable to perform long reorderings, which leads to an important loss in the fluency of the output translation.

On the other hand, the addition of a dictionary when using the union alignment ensures that most of content words are translated, assuring that 'most of the information' is included. This is a typical problem of statistical machine translation systems, which tend to make stupid syntactic or morphological mistakes while still providing a 'fair' message translation. Some examples of translation and one reference for the development set are shown in Table 6.

Translation:	*that what time start ?*
Reference:	*what time does it start ?*
Translation:	*stomach very hurts .*
Reference:	*i have a severe pain in my stomach .*

Table 6: *Samples of translations and reference (dev set)*

Finally, we would like to point out the seemingly inconsistent results of automatic evaluation measures, which demand further research towards finding more robust ways to measure translation performance.

6. Conclusion and further work

The statistical machine translation TALP system has been presented in detail. Description and training details from a parallel corpus have been shown. An evaluation in the framework of Chinese-English supplied task of the IWSLT'04 workshop has been performed. Results have been discussed, addressing the limitations of the system that are highlighted by this challenging translation task.

Future work to improve the system should necessarily tackle the problem of embedded N-grams. One way of treating them would be to extract their translation in a dictionary as it is currently done with embedded words. This would lead to a phrase-based-like approach, but with a reduced set of phrases compared to current approaches.

Moreover, a generalization of the extracted tuples is necessary, for example using classification algorithms or clustering. This could give the system the power of translation unseen tuples adequately, by using the context of 'similar' seen tuples.

And last but not least, techniques to overcome the reordering limitation must be researched, even if that means some big structural change in the translation model based on FST, which proves currently inadequate for pairs of language with different word ordering.

7. Acknowledgements

The authors want to thank Josep Maria Crego and José A. R. Fonollosa (members of the TALP Research Center) for their contribution to this work.

8. References

[1] A. Bonafonte and J. Mariño, "Language modeling using X-grams," *Proc. of the 4th Int. Conf. on Spoken Language Processing, ICSLP'96*, pp. 394–397, October 1996.

[2] Giza++ software, "http://www-i6.informatik.rwth-aachen.de/~och/software/giza++.html."

[3] P. Brown, S. Della Pietra, V. Della Pietra, and R. Mercer, "The mathematics of statistical machine translation," *Computational Linguistics*, vol. 19, no. 2, pp. 263–311, 1993.

[4] S. Vogel, H. Ney, and C. Tillmann, "Hmm-based word alignment in statistical translation," *Proc. of the Int. Conf. on Computational Linguistics, COLING'96*, pp. 836–841, August 1996.

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

[6] ——, "Experiments in word-ordering and morphological preprocessing for transducer-based statistical machine translation," *IEEE Automatic Speech Recognition and Understanding Workhsop, ASRU'03*, November 2003.

[7] F. Casacuberta, "Finite-state transducers for speech input translation," *IEEE Automatic Speech Recognition and Understanding Workhsop, ASRU'01*, December 2001.

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