The TALP Ngram-based SMT System for IWSLT'05

Josep M. Crego, Adrià de Gispert and José B. Mariño

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

Abstract

This paper provides a description of TALP-Ngram, the tuple-based statistical machine translation system developed at the TALP Research Center of the UPC (Universitat Politècnica de Catalunya). Briefly, the system performs a log-linear combination of a translation model and additional feature functions. The translation model is estimated as an N-gram of bilingual units called tuples, and the feature functions include a target language model, a word penalty, and lexical features, depending on the language pair and task.

The paper describes the participation of the system in the second international workshop on spoken language translation (IWSLT) held in Pittsburgh, October 2005. Results on Chinese-to-English and Arabic-to-English tracks using supplied data are reported.

1. Introduction and overview of the system

During the last several years, statistical machine translation (SMT) has gained much attention within the research community. This is mainly due to its relatively easy development in terms of human effort, its robustness in face of non-grammatical input data (such as recognised speech), and its good results against rule-based and transfer-based approaches.

The statistical approach to machine translation is based on the assumption that every sentence t in the target language is a possible translation of a given sentence s in the source language, and the main difference between two translation hypotheses is a probability assigned to each, which is to be learned from a bilingual corpus. The first SMT systems were based on the noisy channel approach on a word-based basis, modeling the translation of a target language sentence t given a source language sentence t as a translation model probability $p(s|t)$ times a target language model probability $p(t)$ [1].

Recently, word-based translation models have been replaced by phrase-based translation models [2, 3], which are estimated from aligned bilingual corpora by using relative frequencies.

On the other hand, according to the maximum entropy framework [4], we can define the translation hypothesis t given a source sentence s, as the target sentence maximizing a log-linear combination of feature functions, as described in the following equation:

$$\hat{t}_1^I = \arg\max_{t_1^I} \left\{ \sum_{m=1}^{M} \lambda_m h_m(s_1^J, t_1^I) \right\} \tag{1}$$

where λ_m correspond to the weighting coefficients of the log-linear combination, and the feature functions $h_m(s,t)$ to a logarithmic scaling of the probabilities of each model.

Following this approach, the translation system described in this paper implements a log-linear combination of one translation model and **four** additional feature models. In contrast with standard phrase-based approaches, our translation model is expressed in *tuples* as bilingual units. Given a word alignment, tuples define a unique and monotonic segmentation of each bilingual sentence, building up a much smaller set of units than with phrases and allowing N-gram estimation to account for the history of the translation process [5, 6]. This approach has its origins in SMT by using finite state transducers [7, 8, 9].

The organization of the paper is as follows. Section 2 describes in detail the tuple n-gram translation model, while section 3 introduces the additional features used in the system. Section 4 provides a brief overview of the decoding tool and search strategy used. Next, sections 5 and 6 report and discuss results on IWSLT'05 Chinese-to-English and Arabic-to-English tracks, respectively. Finally, Section 7 concludes and outlines future research lines.

2. The Tuple N-gram translation model

The tuple N-gram translation model is a language model of a particular language composed by bilingual units which are referred to as tuples. This model approximates the joint probability between source and target languages by using N-grams as described by the following equation:

$$p(s_1^J, t_1^I) = \cdots = \tag{2}$$

$$\prod_{i=1}^{K} p((s,t)_i | (s,t)_{i-N+1}, ..., (s,t)_{i-1}) \tag{3}$$

where $(s,t)_i$ refers to the i^{th} tuple of a given bilingual sentence pair, which is segmented into K tuples. It is important to notice that, since both languages are linked up in tuples, the context information provided by this translation model is bilingual.

Tuples are extracted from a word-to-word aligned corpus according to the following constraints [10]:

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

As a consequence of these constraints, only one segmentation is possible for a given parallel sentence pair and a word alignment. Usually, automatic word-to-word alignments are generated in both source-to-target and target-to-source directions by using GIZA++ [11], and tuples are usually extracted from the union set of alignments. However, in section 5 results are also reported when extracting tuples with the alignment from source-to-target direction. Figure 1 presents a simple example illustrating the tuple extraction process.

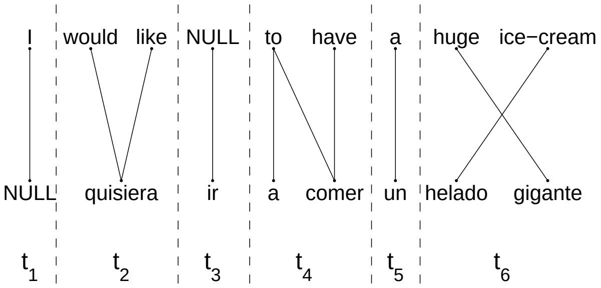

Figure 1: Example of tuple extraction from an aligned bilingual sentence pair.

Once tuples have been extracted, the tuple vocabulary can be pruned by using histogram counts, thus keeping the N most frequent tuples sharing the same source side. Given the reduced size of the supplied IWSLT data, this pruning was not found necessary. Then, the tuple N-gram model can be trained by using any Language Modeling toolkit.

2.1. Tuples with NULL in source

An important issue regarding tuple definition and extraction is the fact that some words linked to NULL end up producing tuples with NULL source sides, as with tuple t_3 from figure 1. Since no NULL is actually expected to occur in translation inputs, this kind of tuple cannot be allowed. Therefore, the target side of the tuple is attached to either the previous or the next tuple in the tuple sequence.

In order to decide to which tuple is attached the '*source-nulled*' tuple, as a baseline option we link the tuple to the following tuple. However, an improved technique has been developed which incorporates IBM-1 probabilities, deciding for the segmentation with higher probability.

This technique incorporates both segmentations (where the target word of the source-nulled tuple is attached to the previous and the next tuple).

In order to score each segmentation, both tuples (next and previous) are taken into account, computing the sum of an IBM-1 weight for each tuple. This weight is computed as follows:

$$\frac{1}{I}\prod_{j=1}^{J}\sum_{i=0}^{I}p_{IBM1}(t^i|s^j)p_{IBM1'}(t^i|s^j) \tag{4}$$

where s and t are the source and target tuple sides, I and J their length in words and $IBM1'$ stands for the reversed IBM model 1. Finally, the sum with the best score defines the best segmentation.

2.2. Embedded words

Another important issue regarding the tuple-based translation model is the existence of embedded words. Given the constraints and the sequentiality defining the tuples, it may happen that a certain amount of single-word translation probabilities are left out of the model. This occurs for those words always appearing embedded into tuples containing two or more words. Consider for example the word “ice-cream” from figure 1. As seen from the figure, “ice-cream” appears embedded into tuple t_6. If a similar situation is encountered for all occurrences of “ice-cream” in the training corpus then no translation probability for an independent occurrence of such word will exist.

To overcome this problem, the tuple N-gram model is enhanced by incorporating 1-gram translation probabilities for all the embedded words detected during the tuple extraction step [9]. These 1-gram translation probabilities are computed from the intersection of both source-to-target and target-to-source alignments.

2.3. Tuple unfolding

When dealing with pairs of languages with very non-monotonic alignments, such as Chinese and English, the sequentiality contraint may lead to an unpractical tuple length and excessive amount of embedded words. In this case, it is more reasonable to allow for a certain reordering in the training data. This means that the tuples are broken into smaller tuples, and these are sequenced in the order of the target words.

In order not to lose the information on the correct order, the decoder performs then a reordered search, which is guided by the N-gram model of the unfolded tuples and the additional feature models. On the other hand, the tuple unfolding process highly reduces the effect of embedded words [12]. Figure 2 shows an example of tuple unfolding compared to the monotonic extraction. The unfolding technique produces a different bilingual N-gram language model with reordered source words.

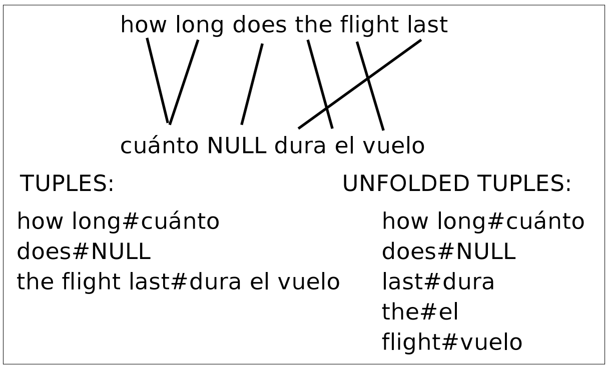

Figure 2: *Example of tuple and unfolded (target-reordered) tuple extraction.*

3. Additional feature models

As additional feature functions to better guide the translation process, TALP incorporates the following models:

- a target language model
- a word penalty model
- a source-to-target lexicon model
- a target-to-source lexicon model

3.1. Target language model

The first of these feature functions is a standard *target language model*, estimated as an N-gram over the target words, as expressed by this equation:

$$p_{LM}(t_k) \approx \prod_{n=1}^{k} p(w_n | w_{n-2}, w_{n-1}) \qquad (5)$$

where t_k refers to the partial translation hypothesis and w_n to the n^{th} word in it.

Although this model could be trained from a larger monolingual data set, this has not been done for IWSLT'05 experiments, which use as target text the same amount of data used as parallel text. As with the tuple translation model, the SRI Language Modeling toolkit was used.

Usually, this feature function is accompanied by a *word penalty model*. This model introduces a sentence length penalty in order to compensate the system's preference for short target sentences, caused by the presence of the previous target language model. This penalization depends on the total number of words contained in the partial translation hypothesis, and it is computed as follows:

$$p_{WP}(t_k) = exp(\text{number of words in } t_k) \qquad (6)$$

where, again, t_k refers to the partial translation hypothesis.

3.2. Lexicon models

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

$$p_{IBM1}((t,s)_n) = \frac{1}{(I+1)^J} \prod_{j=1}^{J} \sum_{i=0}^{I} p(t_n^i | s_n^j) \qquad (7)$$

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

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

4. N-gram based Decoding

For decoding given the combination of models presented above, we used MARIE, a decoder implemeting a beam search strategy with distortion (or reordering) capabilities developed at the TALP Research Center [13]. For efficient pruning of the search space, several pruning techniques are used, such as:

- *Threshold pruning*: Hypotheses with lower scores than a certain threshold are eliminated.
- *Histogram pruning*: Only the K-best ranked hypotheses are kept at each search list of states (covering the same words of the input sentence).
- *Hypothesis recombination*: At each step of the search, two or more hypotheses are recombined if they agree in both the present tuple and the tuple N-gram history.

When allowing for reordering, the pruning strategies are not enough to reduce the combinatory explosion without an important loss in translation performance. For this purpose, two reordering strategies are used:

- A distortion limit (m): Any source word (phrase or tuple) is only allowed to be reordered if it does not exceed a distortion limit, measured in words.
- A reordering limit (j): Any translation path is only allowed to perform j reordering jumps.

The use of reordering strategies implies a necessary trade-off between quality and efficiency. Further details of these reordering strategies are given in the experiments reported in section 5.

5. IWSLT'05 Experiments

The presented system has been evaluated in the framework of the second International Workshop on Spoken Language Translation (IWSLT'05). In the workshop, an Evaluation Campaign has been conducted for five translation directions. Moreover, four different tracks per direction have been proposed, namely using only the supplied corpus (supplied) and allowing the use of NLP tools, additional public data and additional proprietary data, respectively.

TALP has participated in the Chinese-to-English and Arabic-to-English supplied tracks. Next, details on these experiments are presented.

5.1. Corpus and preprocessing

Preprocessing is an optional and language-dependent stage, according to the availability of resources. A minor preprocessing step was carried out in both translation tasks. As evaluation is performed without punctuation marks, we experimented with training without punctuation, but this was discarded as results were equal to or worse than leaving punctuation until a final output post-processing.

Tables 1 and 2 show the main statistics of the supplied data, namely number of sentences, words, vocabulary, and maximum and average sentence lengths for each language, respectively. A development set of 1006 sentences was also supplied, together with 16 reference English translations (CSTAR03 plus IWSLT04 test sets). Note that $Arabic'$ refers to the statistics of the re-tokenized Arabic corpus as explained in Section 5.2.

5.2. Training details

The training of the system is comprised of several stages with the objective of building the four models used by the system.

supplied	sent.	words	voc.	Lmax	Lavg
Train set					
Chinese	20,000	176,199	8,687	68	8.81
English		182,257	7,316	75	9.11
Development set					
Chinese	1006	7,309	1,384	62	7.27
Test set					
Chinese	506	3,743	963	56	7.4

Table 1: *Chi-Eng supplied corpus statistics. There are 257 and 155 unseen words in the dev and test sets.*

supplied	sent.	words	voc.	Lmax	Lavg
Train set					
Arabic	20,000	131,712	25,186	50	6.59
Arabic'		180,477	15,956	70	9.02
English		182,257	7,316	75	9.11
Development set					
Arabic	1006	5,291	2,353	50	5.26
Arabic'		7,217	1,884	68	7.17
Test set					
Arabic	506	2,607	1,387	46	5.13
Arabic'		3,632	1,179	57	7.15

Table 2: *Ara-Eng supplied (and re-tokenized) corpus statistics. There are 303 and 146 unseen words in the re-tokenised dev and test sets.*

The histograms in Figure 3 show the number of tuples found in the corpus over the tuple size for both translation tasks.

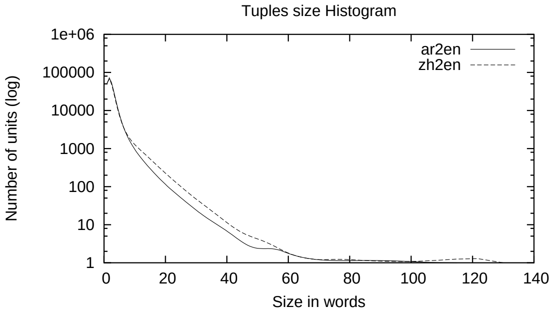

Figure 3: *Number of tuples found in training over the tuple size for each translation task.*

The preprocessing stage was only performed on the Arabic side of the corpus, and apart from standard punctuation marks, it aims at separating prefixes (such as the article) that would highly increase the vocabulary size if considered as parts of words. In detail, we produce a hard separation of all words starting with ال and بال (as ب + ال), in order to separate articles from words. Note that this process is neither guided by tagging information (it does not use any tagging software) nor complete (several other Arabic particles are usually attached to words). However, it already produces a significative vocabulary reduction, leading to improved performance.

During word alignment, IBM model 1 tables are used directly to compute the lexicon feature. Finally, in order to learn the target and the tuples language models we used SRILM [14]. All models were learnt using interpolation of higher and lower order n-grams with Knesser-Ney [15] smoothing.

5.3. Development work

Several configurations were tested on the development set optimizing BLEU, namely baseline and three alternatives. Results are shown in table 3.

The **baseline** configuration system is built using:

- The union alignment [11] to extract unfolded tuples and the intersection to solve embedded words.
- All source-nulled tuples are linked the the target word of the next tuple.
- The order of the target and the translation Ngram language models is set to 4 and 3, respectively.
- The reordering parameters of the decoder are fixed to $m = 5$ and $j = 3$ for the Chinese-to-English task, and $m = 3$, $j = 3$ for the Arabic-to-English task. This settings suppose a necessary trade-off between quality and efficiency. As reordering is not so critical in the Arabic task and does not produce any big improvement in quality, a smaller distortion distance limit is used.
- The unfolding procedure detailed in section 2.3.

zh2en	BLEU	NIST	mWER	PER
baseline	0.358	6.67	46.77	39.75
4grBM	0.357	7.03	47.45	40.01
NULLibm	0.365	7.50	47.47	40.16
sAt	0.384	7.45	48.63	41.77
ar2en	**BLEU**	**NIST**	**mWER**	**PER**
baseline	0.554	9.12	30.78	26.76
4grBM	0.553	9.09	30.87	26.83
NULLibm	0.554	9.13	30.64	26.70
sAt	0.573	7.84	30.94	28.12

Table 3: *Evaluation results (development set) when optimizing BLEU in both translation tasks.*

Three alternative configurations have been studied. In **4grBM** the order of the translation Ngram language model is increased to 4. In **NULLibm**, the 4grBM configuration is improved by solving source-nulled tuples following the method described in 2.1. Finally, the NULLibm configuration is further extended in **sAt**, where the source-to-target alignment is also used for tuple extraction (together with the union). This way, the tuple language model is learnt from the concatenation of those tuples extracted from the union alignment and those from the source-to-target alignment.

Even though the use of **4grBM** does not seem to produce any change in quality, we decided to include this in our experiments based on previous development work with a different BLEU score implementation (used in IWSLT'04), where significant improvements were obtained when compared to the **baseline**.

In the Chinese-to-English task (zh2en), the best BLEU results are obtained when using the **sAt** configuration, which is built using all the additional features (4-grams in the bilingual LM, solving source-nulled tuples using the IBM-1 lexicon model, and making use of the additional source-to-target alignment).

On the contrary, the Arabic-to-English task (ar2en) does not seem to take advantage from any of the additional features except for the introduction of the source-to-target alignment in **sAt**.

We observe a clear contradiction regarding BLEU and the other scores when adding the source-to-target alignemnt **sAt** in both translation tasks (see the increase in mWER for zh2en and decrease in NIST for ar2en). Trying to understand this situation, we performed a mWER optimization using two configurations, **sAt** and **NULLibm**. Results are shown in table 4.

zh2en	BLEU	NIST	mWER	PER
NULLibm	0.349	5.94	46.01	39.72
sAt	0.368	5.70	45.03	39.28
ar2en	**BLEU**	**NIST**	**mWER**	**PER**
NULLibm	0.551	9.13	30.63	26.70
sAt	0.546	9.05	30.91	27.15

Table 4: *Evaluation results (development set) when optimizing mWER in both translation tasks.*

When optimizing mWER (see Table 4), the Chinese-to-English task shows a clear improvement when using **sAt** (measured in mWER and BLEU) at the cost of a lower NIST scores. While in the Arabic-to-English task, a very slight improvement is achieved (measured in mWER) while worst scores are obtained for both BLEU and NIST.

To outline this contradiction, four different test set runs were submitted for each language pair, namely the optimizations of BLEU and mWER for both the **NULLibm** and **sAt** configurations. As primary submission, we selected the **sAt** configuration with weights optimized maximizing BLEU. The secondary submission consists of the **NULLibm** configuration with weights optimized minimizing mWER.

The optimizations were performed using an in-house developed tool based on the simplex method [16].

5.4. Test set results

The evaluation scores of the TALP-Ngram system (primary and secondary submissions), obtained in both translation tasks are shown in table 5. In Table 6 different implementations of mWER and PER scores are used for development and test sets.

zh2en	BLEU	NIST	mWER	PER
primary	0.444	8.40	48.23	40.79
secondary	0.434	6.89	47.24	39.49
ar2en	BLEU	NIST	mWER	PER
primary	0.533	6.541	39.93	36.77
secondary	0.514	8.467	38.78	33.76

Table 5: *Results obtained by the TALP-Ngram system in both translation tasks. Two runs were submitted for each task.*

As it can be observed, the BLEU and NIST scores are correlated for both dev and test sets in the zh2en task, both improving in the primary run. However, they are incorrelated for both dev and test sets in the ar2en task.

6. Discussion

When studying the test results, we can note that the Chinese test set seems to be 'easier' to translate than the development (obtaining higher scores), whereas the effect is opposite in the case of Arabic. This behaviour could easily be explained by the nature of the data. However, when comparing the two TALP systems which competed in the same tracks and under the same conditions (TALP-Ngram and TALP-Phrase [17]), a surprisingly different behaviour between development and test can be found. Regarding development results, the TALP-Ngram system improves the performance of the TALP-Phrase system (table 6) in the Chinese-to-English task ($0.384 > 0.373$), while it achieves the same score in the Arabic-to-English task ($0.573 \approx 0.572$), both measured in BLEU. However, regarding the test set, the TALP-Ngram system is clearly beaten by the TALP-Phrase system in both tasks ($0.444 < 0.452$ in Chinese-to-English, and $0.533 < 0.573$ in Arabic-to-English). Experiments have been conducted in order to find out the reason explaining this different behaviour.

The results obtained by both systems are shown (primary submissions are only discussed) in table 6.

System (zh2en)	BLEU	NIST	mWER	PER
Ngram (dev)	0.384	7.45	48.63	41.77
Ngram (test)	0.444	8.40	48.23	40.79
Phrase (dev)	0.373	6.90	46.54	39.01
Phrase (test)	0.452	7.97	45.91	37.96
System (ar2en)	BLEU	NIST	mWER	PER
Ngram (dev)	0.573	7.84	30.94	28.12
Ngram (test)	0.533	6.54	39.93	36.77
Phrase (dev)	0.572	9.87	30.50	26.39
Phrase (test)	0.573	9.33	35.00	30.30

Table 6: *Results obtained by the two TALP systems participating in IWSLT'05 (on dev and test sets) in both translation tasks. Note that mWER and PER scores are computed using different implementations in development and test. Test scores are all computed using the IWSLT'05 official scores.*

The same decoder (MARIE [13]), optimization tool [16] and additional models (target 4-gram LM, IBM-1 lexicon model, reordering model, word penalty) were used in both systems. Furthermore, the same additional tokenization was performed on the Arabic source side of the corpus. Differences are found on the bilingual units used (tuples versus phrases), their translation models (Ngram LM versus relative frequencies), and two additional models used by the TALP-Phrase system (a phrase penalty and a relative frequency translation model computed from target to source).

Comparing the model weights obtained by both systems after the optimization (shown in table 7), we can see how the TALP-Ngram system does not make use of any of the IBM-1 lexicon models in the Arabic-to-English translation task.

	zh2en		ar2en	
Model	Phrase	Ngram	Phrase	Ngram
TM	5.23	2.00	4.40	1.05
WP	1.90	1.86	2.23	0.51
RM	0.17	0.09	1.27	0.17
IBM1	0.07	0.08	0.03	0.02
IBM1'	0.24	0.41	0.32	**0.03**
TM'	1.15	-	1.00	-
PP	1.06	-	0.42	-

Table 7: *Model weights used by the TALP Phrase and Ngram systems in primary runs. Bilingual model weights are always set to 1, and the rest of weights are (from top to bottom): target LM, word penalty, reordering model, IBM-1 lexicon models (source-to-target and target-to-source), target-to-source bilingual model (computed using relative frequencies) and phrase penalty.*

For cross-validation, the development was divided into two subsets (dev1, ie. 500 CSTAR'03 sentences and dev2, ie. 506 IWSLT'04 sentences), and optimizations were performed with each dev subset, evaluating on the test set. Results are shown in table 8. As it can be seen, the tendency remains the same when optimizing with dev1, dev2 or fulldev, leading to a surprising decrease in Arabic-to-English performance in the test set.

task	optim.	eval.	system	BLEU
zh2en	dev1	dev2	Ngram	0.376
	dev1	dev2	Phrase	0.372
	dev1	test	Ngram	0.453
	dev1	test	Phrase	0.441
zh2en	dev2	dev1	Ngram	0.392
	dev2	dev1	Phrase	0.391
	dev2	test	Ngram	0.446
	dev2	test	Phrase	0.453
ar2en	dev1	dev2	Ngram	0.556
	dev1	dev2	Phrase	0.548
	dev1	test	Ngram	0.523
	dev1	test	Phrase	0.566
ar2en	dev2	dev1	Ngram	0.582
	dev2	dev1	Phrase	0.574
	dev2	test	Ngram	0.535
	dev2	test	Phrase	0.562

Table 8: *BLEU score computed over different sets optimizing with different dev sets.*

A further comparison was performed in terms of the units used when translating development and test sets. The experience of the authors is that in many cases, translation errors are related to tuples with NULL in the target side. Therefore, table 9 studies the number of these units used in translating the dev and test sets. However, no relevant difference can be observed. As the development size is approximately double the test size, the same happens with tuples to NULL.

zh2en	tpl2NULL	%1gr
dev	1396	38.9
test	643	36.4
ar2en	**tpl2NULL**	**%1gr**
dev	1554	38.4
test	833	36.9

Table 9: *Number of translation units used with NULL in the target side, and the percentage of these units translated as 1grams.*

The percentage of these tuples which are unigrams (uncontextual, and usually leading to errors) is also similar. Therefore, no conclusion can be drawn.

Table 10 shows the number of tuples used as 1-grams, 2-grams, 3-grams and 4-grams by the TALP-Ngram in both translation tasks regarding development and test sets. Again, no special difference is found in the Arabic test, being the figures approximately the double as with the dev set.

zh2en	1gr	2gr	3gr	4gr
dev	2818	2786	789	325
test	1286	1396	486	286
ar2en	**1gr**	**2gr**	**3gr**	**4gr**
dev	2275	2792	1013	495
test	1165	1426	524	255

Table 10: *Ngrams used by the TALP-Ngram system in both translation tasks when translating the development and test sets.*

		zh2en	ar2en
system	set	words	words
Ngram	dev	5581	4983
Phrase	dev	5325	5647
Ngram	test	2913	2421
Phrase	test	2810	2750

Table 11: *Number of words output by the TALP-Ngram and TALP-Phrase systems in both translation tasks when translating the development and test sets.*

Additionally, Table 11 presents the number of output words produced by the TALP-Ngram and TALP-Phrase systems, as our experience is that differences in length may produce differences in BLEU score. However, whereas the TALP-Phrase always produces shorter outputs in Chinese-to-English, the behaviour is opposite in Arabic, without inconsistencies between dev and test.

Therefore, we have not yet found a reason to explain the difference in performance regarding development and test sets (perhaps just an artifact of the corpora?). Further research should be conducted to explain such a behaviour.

Another point of discussion is the optimization procedure. It seems to be a weak point of current SMT systems. The use of optimization algorithms like simplex [16] showed to be effective when applied over spaces with two or three dimensions. Current SMT systems are built using more than four additional models which have to be optimized at the same time.

Optimization over spaces with many dimensions conveys a lot of local maxima, which are typically solved through a limited number of restarts. This situation makes the final optimization highly dependent of the initial point, which is very often chosen almost randomly.

7. Conclusion and further work

In this paper we have presented the TALP Ngram-based statistical machine translation system (TALP-Ngram). Description and training details have been shown for the IWSLT'05 evaluation workshop, consisting of a Chinese-to-English and an Arabic-to-English translation tasks.

Two configurations have been submitted for each translation task in order to outline the contradiction between BLEU and mWER, and the contradiction between BLEU and NIST (clearly unexpected as both account for a weighted match of word Ngrams).

Results have been presented, highlighting the strong differences in behaviour found between the development and test sets, when compared to another participating system (TALP-Phrase).

Future work is necessary to overcome problems such as the occurrence of NULL words in the translation units and the optimization process with high dimensional spaces.

8. Acknowledgements

This work has been partially funded by the European Union under the integrated project TC-STAR - Technology and Corpora for Speech to Speech Translation -(IST-2002-FP6-506738, `http://www.tc-star.org`), by the Spanish Government under grant TIC2002-04447-C02 (ALIADO project), by the Dep. of Universities, Research and Information Society (Generalitat de Catalunya) and by the Universitat Politècnica de Catalunya under grant UPC-RECERCA.

The authors want to thank Marta Ruiz Costa-jussà (member of the TALP Research Center) for her valuable contribution to the comparison with the TALP-Phrase system.

9. References

[1] 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.

[2] R. Zens, F. Och, and H. Ney, “Phrase-based statistical machine translation,” in *KI - 2002: Advances in artificial intelligence*, M. Jarke, J. Koehler, and G. Lakemeyer, Eds. Springer Verlag, September 2002, vol. LNAI 2479, pp. 18–32.

[3] P. Koehn, F. Och, and D. Marcu, “Statistical phrase-based translation,” *Proc. of the Human Language Technology Conference, HLT-NAACL’2003*, May 2003.

[4] A. Berger, S. Della Pietra, and V. Della Pietra, “A maximum entropy approach to natural language processing,” *Computational Linguistics*, vol. 22, no. 1, pp. 39–72, March 1996.

[5] R. E. Banchs, J. M. Crego, A. de Gispert, P. Lambert, and J. B. Mariño, “Statistical machine translation of euparl data by using bilingual n-grams,” *Proc. of the ACL Workshop on Building and Using Parallel Texts (ACL’05/Wkshp)*, pp. 67–72, June 2005.

[6] J. Mariño, R. Banchs, J. Crego, A. de Gispert, P. Lambert, M. R. Costa-jussà, and J. Fonollosa, “Bilingual n–gram statistical machine translation,” *Proc. of the MT Summit X*, September 2005.

[7] E. Vidal, “Finite-state speech-to-speech translation,” *Proc. of 1997 IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, pp. 111–114, 1997.

[8] 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.

[9] ——, “Talp: Xgram-based spoken language translation system,” *Proc. of the Int. Workshop on Spoken Language Translation, IWSLT’04*, pp. 85–90, October 2004.

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

[11] F. Och and H. Ney, “Improved statistical alignment models,” *38th Annual Meeting of the Association for Computational Linguistics*, pp. 440–447, October 2000.

[12] J. M. Crego, J. Mariño, and A. Gispert, “Reordered search and tuple unfolding for ngram-based smt,” *Proc. of the MT Summit X*, September 2005.

[13] ——, “An ngram-based statistical machine translation decoder,” *Proc. of the 9th European Conference on Speech Communication and Technology, Interspeech’05*, September 2005.

[14] A. Stolcke, “Srilm - an extensible language modeling toolkit,” *Proc. of the 7th Int. Conf. on Spoken Language Processing, ICSLP’02*, September 2002.

[15] S. Chen and J. Goodman, “An Empirical Study of Smoothing techniques for Language Modeling,” in *Proceedings of 34th ACL*, San Francisco, July 1996, pp. 310–318.

[16] J. Nelder and R. Mead, “A simplex method for function minimization,” *The Computer Journal*, vol. 7, pp. 308–313, 1965.

[17] M. R. Costa-jussà and J. Fonollosa, “Tuning a phrase-based statistical translation system for the iwslt 2005 chinese to english and arabic to english tasks,” *IWSLT05*, October 2005.