On the use of statistical machine- translation techniques within a memory-based translation system (AMETRA)

Daniel Ortiz[2]
Ismael García-Varea[1]
Francisco Casacuberta[2]
Antonio Lagarda[2]
Jorge González[2]
[1] Departamento de Informática
Universidad de Castilla-La Mancha
02071 Albacete, SPAIN
ivarea@info-ab.uclm.es

[2]Instituto Tecnológico de Informática
Departamento de Sistemas Informáticos y Computación
Universidad Politécnica de Valencia
46071 Valencia, SPAIN
dortiz@iti.upv.es
fcn@iti.upv.es

Abstract

The goal of the AMETRA project is to make a computer-assisted translation tool from the Spanish language to the Basque language under the memory-based translation framework. The system is based on a large collection of bilingual word-segments. These segments are obtained using linguistic or statistical techniques from a Spanish-Basque bilingual corpus consisting of sentences extracted from the Basque Country's of£cial government record. One of the tasks within the global information document of the AMETRA project is to study the combination of well-known statistical techniques for the translation of short sequences and techniques for memory-based translation. In this paper, we address the problem of constructing a statistical module to deal with the task of translating segments. The task undertaken in the AMETRA project is compared with other existing translation tasks, This study includes the results of some preliminary experiments we have carried out using well-known statistical machine translation tools and techniques.

1 Introduction

Over the last few years has became more and more popular the integration of different techniques in the development of machine translation systems. Currently, most of the existing commercial systems make use of the best parts of different approaches to obtain better results and £nally better products.

The aim of the AMETRA project is to include some statistical translation techniques, which have been successfully applied in speci£c domain tasks to improve the results of a Spanish-Basque computer-assisted translation system, which uses a memory-based approach.

The success of a statistical machine translation system relies on the availability of a large bilingual corpus to be used to train different translation and language models. Thus, is specially important the quality of such a corpus in terms of complexity. Ideally, the corpus should be perfectly split into sentences, be free of noise and errors and be free as possible of incorrect translations. In practice, this is not the usually the case. New corpora usually require substantial preprocessing as is the case with our corpus. We show how the statistical techniques can be succesfully applied and how the statistical and the translation memory approaches can be combined to a translation of Spanish to Basque.

2 Statistical Machine Translation (review)

The goal of the translation process in *statistical machine translation* (SMT) can be formulated as follows: A source language string $f_1^J = f_1 \dots f_J$ is to be translated into a target language string $e_1^I = e_1 \ldots e_I$. In the experiments reported in this paper, the source language is Spanish and the target language is Basque. Every target string is considered as a possible translation for the input. If we assign a probability $Pr(e_1^I|f_1^J)$ to each pair of strings (e_1^I, f_1^J), then according to Bayes' decision rule, we have to choose the target string that maximizes the product of the target language model $Pr(e_1^I)$ and the string translation model $Pr(f_1^J|e_1^I)$.

Many existing systems for statistical machine translation make use of a special way of structuring the string translation model as proposed by (Brown et al., 1993): The correspondence between the words in the source and the target string is described by alignments that assign one target word position to each source word position. The lexicon probability $p(f|e)$ of a certain target word e occurring in the target string is assumed to depend basically only on the source.

These alignment models are similar to the concept of Hidden Markov models (HMM) in speech recognition. The alignment mapping is $j \rightarrow i = a_j$ from source position j to target position $i = a_j$. The alignment a_1^J may contain alignments $a_j = 0$ with the 'empty' word e_0 to account for source words that are not aligned to any target word. In (statistical) alignment models $Pr(f_1^J, a_1^J|e_1^I)$, the alignment a_1^J is introduced as a hidden variable.

Typically, the search is performed using the so-called maximum approximation:

$$\begin{aligned}\hat{e}_1^I &= \arg\max_{e_1^I} \left\{ Pr(e_1^I) \cdot \sum_{a_1^J} Pr(f_1^J, a_1^J|e_1^I) \right\} \\ &\approx \arg\max_{e_1^I} \left\{ Pr(e_1^I) \cdot \max_{a_1^J} Pr(f_1^J, a_1^J|e_1^I) \right\}\end{aligned}$$

The search space consists of the set of all possible target language strings e_1^I and all possible alignments a_1^J.

In this work, we used IBM Model 1 and IBM Model 4 (Brown et al., 1993) as translation models. With respect to the language models, we used a trigram language performed using the Good-Turing estimate and smoothed by the Katz technique. Finally, as a decoding algorithm we used a stack-based decoder which is outlined in more detail in the next section.

3 Stack-based decoding

The stack decoding algorithm, also called A^* algorithm, was introduced by F. Jelinek in (Jelinek, 1969) the first time. The stack decoding algorithm attempts to generate partial solutions, called *hypotheses*, until a complete sentence is found; these hypotheses are stored in a stack and ordered by their *score*. In our case, this measure is a probability value given by both the translation and the language models. The decoder follows a sequence of steps for achieving an optimal hypothesis:

1. Initialize the stack with an empty hypothesis.
2. Iterate
   (a) Pop h (the best hypothesis) off the stack.
   (b) If h is a complete sentence, output h and terminate.
   (c) Expand h.
   (d) Go to step 2a.

The search is started from a null string and obtains new hypotheses after an expansion process (step 2c) which is executed at each iteration. The expansion process consists of the application of a set of operators over the best hypothesis in the stack. Thus, the design of stack decoding algorithms involves defining a set of operators to be applied over every hypothesis as well as the way in which they are combined in the expansion process. Both the operators and the expansion algorithm depend on the translation model that we use. In our case, we used IBM Model 3 and IBM Model 4.

The operators used in the implementation are those defined in (Berger et al., 1996) and (Germann et al., 2001) for the IBM Model 3 and IBM Model 4.

The expansion we used in each iteration is strongly inspired on the expansion given in (Berger et al., 1996) for the IBM Model 3, and was presented in (Ortiz et al., 2003). This algorithm had been previously adapted for the IBM Model 4, and additionaly has been adapted in this work for the IBM Model 1.

4 The AMETRA corpus

The AMETRA corpus is a bilingual corpus from the Spanish language to Basque language. This corpus was extracted from the Basque Country's official government record, which was segmented into sentences during a previous project. The application of statistical machine translation algorithms over this corpus raises several important difficulties:

- The corpus has several segmentation errors.
- The corpus is often inconsistent. Concretely, the inconsistences are due to the machine transcription process of the corpus itself. For example, the same word appears sometimes with all its symbols in upper case, and sometimes in lower case; or for the Spanish language, the same word is sometimes accentuated and other times not, etcetera.
- The corpus presents a high degree of non-monotonicity.

In addition, a lot of names, numbers and dates appear, enormously increasing the size of the vocabularies.

These problems can be partially solved by carrying out a corpus preprocessing. So far, the preprocessing consisted of filtering out punctuation marks.

Both vocabularies are too big in relation to the number of available sentences (even after a preprocessing step). The Basque vocabulary is particularly enormous in this sense. Table 1 shows statistics of the unpreprocessed corpus (the vocabulary size of not preprocessed corpus is shown in parenthesis). The sentences with sixty or more words were not taken into account because we considered them to be paragraphs.

Table 1 also shows the differences between the mean length of the sentences for both languages, this shows the "agglutinative character" of the Basque language in relation to the to the Spanish language. Table 1 also shows the relation between the number of available sentences inf the corpus and the size of the Basque vocabulary. Figure 1 shows a histogram of the Basque sentence length for the preprocessed corpus. A more extensive study shows that whole paragraphs appeared frequently in a single line.

4.1 Comparison with other well known tasks

Before describing the process of splitting sentences into segments, and presenting the results produced by the stack-based decoder, it might be interesting to find out how complex is the AMETRA corpus. A good way to do this is to compare it with other widely studied tasks such as HANSARDS, VERBMOBIL or EUTRANS-I.

First of all, the corpus must be prepared to make a translation experiment. For AMETRA corpus, such an experiment requires a a previous shuffling of the corpus because strong relations exist between consecutive sentences, due to its alphabetical ordering. After the shuffling, the last 1000 sentences were used for test purposes and all the previous for training. Table 2 shows the statistics of the AMETRA whole sentence preprocessed corpus.

	Spanish	Basque
Sentences	89,420	
Words	2,164,019	1,563,292
Vocabulary	58,797 (93,909)	111,638 (158,155)
Mean sentence length	24.2 (23.8)	17.4 (16.3)

Table 1: Statistics of the whole AMETRA corpus.

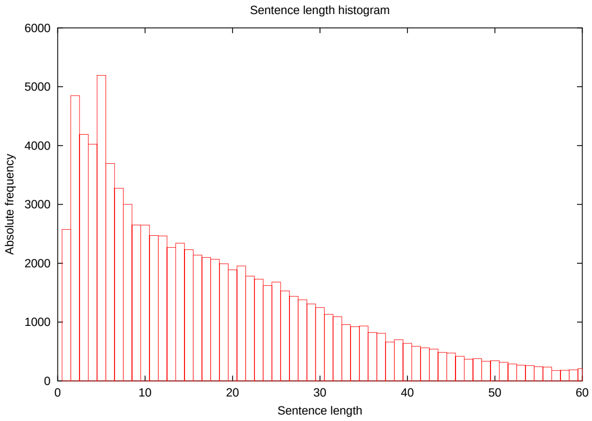

Figure 1: Basque sentence-length histogram.

		Spanish	Basque
Training	Sentences	88,420	
	Words	2,139,491	1,545,454
	Vocabulary	58,515	110,980
	Mean sentence length	24.1	17.4
Test	Sentences	1,000	
	Words	24,528	17,838
	Perplexity (trigrams)	–	367.2

Table 2: Statistics for the AMETRA whole sentence corpus, separated in different training and test subcorpora.

		German	English
Training	Sentences	58,073	
	Words	519,523	549,921
	Vocabulary	7,940	4,673
Test	Sentences	251	
	Words	2,628	2,871
	Perplexity (trigrams)	–	30.5

Table 4: Statistics of the Verbmobil task.

Once the corpus was trained, those sentences whose length were twelve or below were extracted, due to the high complexity of the search process using stack-based decoding algorithms (see (Ortiz et al., 2003)). We obtained twelve subsets of the test corpus, labeled with the symbol t followed by the sentence length, to which the stack-based decoders were applied.

Table 3 shows the measures *WER*[1] (Word Error Rate) and *PER*[2] (Position independent Error Rate) of translation quality for the subsets, as well as the search error rate[3] and the number of translated segments for every subset.

The *PER* and *WER* measures in Table 3, show the high complexity of the AMETRA corpus. However, the search error rate is reasonably low and seems to be related to the number of translations of every source word (also referred as the W parameter within the stack-based decoder, see (Ortiz et al., 2003)). We can state that the search process was carried out correctly, but over a very complex and even badly estimated model due to the negative characteristics of the corpus.

Let's see the big difference between the *WER* and *PER*'s values, which is typical in tasks like VerbMobil (see (Wahlster, 2000)). VerbMobil is related to the tourist domain, and the translation is made from German to English (see Table 4 for some statistics).

Since the *PER* measure does not take word order into account, is appropriate for those tasks where the word ordering between the source and target languages is very different. AMETRA is one of those tasks.

We can also show the statistics of the Hansards task, Hansards task consists of debates in the Canadian Parliament, where French and English are the official languages. It is a well known task and is also very difficult for machine translation, see Table 5.

		French	**English**
Training	Sentences	1,470,473	
	Words	24,338,195	22,163,092
	Vocabulary	100,269	78,332
Test	Sentences	5,432	
	Words	97,646	88,773
	Perplexity (trigrams)	–	179.8

Table 5: Statistics of the Hansards' task.

Translation results were obtained for the Hansards task in (Ortiz et al., 2003) with stack-based decoders and the *WER* measure was never lower than 51 points. However AMETRA is even more complex than Hansards due to the small number of training sentences in relation to the large vocabularies of the languages.

Also note the high perplexity of the AMETRA task (see Table 2) in relation to Hansards.

Perhaps it would be interesting to ask ourselves when machine translation can be successfully applied. The EuTrans-I task, commonly known as the *Traveler task*, is a nice example of a sufficiently simple task that can be translated by a machine, in contrast with AMETRA and Hansards. The EuTrans-I task consists of a semi-automatically generated Spanish–English corpus. The domain of the corpus consists of a human-to-human communication situation at a reception desk of a hotel. The statistics of such a corpus are shown in Table 6. A mean *WER* measure that is lower than 10 points can be achieved without too much computational cost and with no preprocessing step.

Table 7 contains additional data about the obtained language models for the tasks AMETRA, Hansards, VerbMobil and EuTrans-I. The table shows the bigrams and trigrams that appear only once in the training corpus (in percentages). It also shows the number of unseen bigrams and trigrams in the

[1] Defined as the minimum number of insertions, substitutions and deletions that must be done to turn the generated translation into the reference sentence.

[2] Unlike the *WER*, *PER* measure does not take into account the position of the words in either the target or the reference sentence.

[3] We say a search error ocurrs if the sentence generated by the translator is different than the reference sentence and has a worse score.

AMETRA	**WER**	**PER**	**Search errors(%)**	**# of sentences**
t1	65.0	65.0	5,2	19
t2	84.8	84.8	3,4	29
t3	63.2	62.2	2.5	40
t4	80.7	74.4	6.8	58
t5	71.7	67.1	10.2	39
t6	62.7	57.7	14.2	35
t7	69.4	59.2	25.0	20
t8	59.4	53.1	10.0	30
t9	72.2	56.5	9.5	21
t10	77.0	66.3	8.6	23
t11	74.1	68.6	0	21
t12	73.9	66.3	9	11
mean	71.4	65.9	8.3	

Table 3: Translation results for the AMETRA whole-sentence corpus.

		Spanish	**English**
Training	Sentences	10,000	
	Words	97,131	99,292
	Vocabulary	686	513
	Mean sentence length	9.7	9.9
Test	Sentences	2,996	
	Words	35,023	35,590
	Perplexity (trigrams)	–	3.6

Table 6: Statistics of the EuTrans-I task

test corpus [4]. Obviously AMETRA has the most variability and also the highest test corpus perplexity.

Another way of placing AMETRA in the machine translation framework is to compare it with the task in (Al-Onaizan et al., 1999), this task is a Czech-English corpus that was trained and translated. There is a certain paragraph of the document describing the Czech language which says: “In the corpus there are 72 000 word forms in the Czech part versus 31 000 forms in English”, due to the multiplicity of cases, numbers, genders, etc that the Czech language has. This situation is similar to the AMETRA corpus. And the similarities go further because the above mentioned task does not have a great number of training sentences (only 51 000).

5 Corpus segmentation and translation results

The AMETRA project deals with memory-based computed-assisted translation. The database of the systems consists in a large collection of short, bilingual word sequences (*segments*). Statistical techniques can help the process of extracting the bilingual segments from the AMETRA corpus:

5.1 Features of the segmented training and test corpus

We have performed the following sequence of steps over the whole corpus in order to obtain the training and test subcorpus of segments, which will be used to carry out the translation experiments for the segments:

1. A training of the whole partially-preprocessed corpus, using the *GIZA++* tool was carried out. IBM Model 5 was obtained.

2. The best word-alignments in the training set were computed using the *GIZA++* tool and the trained IBM Model 5.

3. The training corpus was segmented according to the following criterion: *A bilingual segment is composed by the shortest sequence of source words and the shortest sequence of target words in such a way that no source words can be aligned with target words that are not in the associated sequence of target words and no target words can be aligned with source words that are not in the associated sequence of source words.*

From the set of bilingual segments, the last 2714 segments were selected for testing purposes and all the previous sentences for training a new translation model. Once again the tool *GIZA++* was used to

[4] For Hansards we give the data corresponding to the use for training purposes of the first 128,000 sentences from the original training corpus.

	% Trig. = 1	% Bigr. = 1	% Unseen Trig.	% Unseen Bigr.	Perplexity
AMETRA (Basque)	85.6	73.2	73.0	37.0	367.2
Hansards (English)	81.4	65.7	53.2	23.3	179.8
VerbMobil (English)	67.6	52.9	41.6	22.7	30.5
EuTrans-I (English)	45.2	35.7	13.6	8.4	3.6

Table 7: Language model statistics for four different translation tasks

		Spanish	Basque
Training	Sentences	229,700	
	Words	2,065,217	1,482,792
	Vocabulary	57,837	110,757
	Mean sentence length	8.9	6.4
Test	Sentences	2,714	
	Words	23,662	17,173
	Perplexity (trigrams)	–	323.2

Table 8: Statistics of the AMETRA segmented corpus.

carry out the training, using the same set of parameters as in step 1. The training of a trigram language model was done by using the *SRILM* toolkit.

Table 8 shows the statistics of the segmented AMETRA corpus, yet divided for training and test purposes. The language model perplexity is also given, and its comparison with the perplexity of other tasks is interesting (see the next subsection). The shuffling of the corpus divides the language model perplexity by two units approximately.

Figure 2 shows how the segmentation has affected the sentence length in the new segmented corpus.

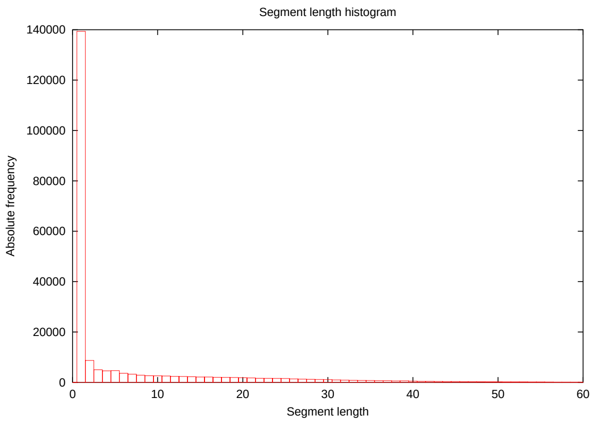

Figure 2: Euskera's segment length histogram.

5.2 Translation segment results

Table 9 shows the translation results for the AMETRA segments.

It's very important to point out that the *WER* and *PER* measures are only valuable if reliable reference sentences are given. In our case the corpus already presented a high level of noise, and is underwent a complex transformation process (it was divided into segments) introducing additional noise. Therefore an increase in the number of incorrect reference sentences is expected.

We observed high values for the *WER* and *PER* measures and an increase in the search error rate. The use of only one sentence as translation reference might be related with these values, since the references are not always trustworthy. Table 10 shows some examples of wrong reference translations. All of them were due to these segmentations errors.

We also observed moderate mean values for the *WER* and *PER* and for the search error rate. However, this is mainly due to the great number of segments having a length equal to one, because we calculated weighted means.

These considerations oblige us to consider the results that appear in the table with a certain amount caution.

5.3 Training and translation with IBM model 1

Due to the problems that the AMETRA corpus has (see section 4), we tried to use the IBM Model 1 in a new translation experiment. IBM Model 1 is the simplest IBM Model, and we supposed that it would perform better in a high noise situation, which is the case of the AMETRA corpus.

The results in Table 11 were obtained using the same segmented corpus as the one presented in Ta-

AMETRA	**WER**	**PER**	**Search err.(%)**	**# of segments**
t1	20.6	19.9	0	711
t2	58.9	58.2	19.0	110
t3	64.2	57.1	16.6	72
t4	77.1	69.1	28.1	64
t5	69.9	59.6	26.3	38
t6	83.5	64.4	37.1	35
t7	70.2	61.4	29.0	31
t8	66.0	54.7	23.3	30
t9	75.9	61.7	40.0	25
t10	76.6	65.8	38.4	26
t11	77.7	64.3	32.0	25
t12	81.8	64.4	57.1	14
mean	40.1	36.3	10.6	

Table 9: Translation results for the segments of the AMETRA task

Spanish	**Basque**
El movimiento de las piezas	Piezen
Fabricación de cales y yesos	Kare eta
Los estancos del Territorio Histórico	estankoetan
teatrales , musicales , coreográficas , audiovisuales	koreografia ,
Los certificados en calidad de visto bueno	Ziurtagiriak
IMPORTANTE : Van a utilizarlo como usuarios	GARRANTZITSUA :
Intereses imposiciones a plazo	Eperako
Calificación : Se califica	Kalifikapena :

Table 10: Some examples of wrong translation references extracted from the AMETRA corpus.

ble 9 but using the IBM Model 1 as the translation model[5].

The use of the IBM Model 1 introduces a slight improvement in relation to the results obtained with IBM Model 4. We attribute it to the noise of the corpus that we have mentioned above. The IBM Model 1 does not care about the correct ordering of the target words; however, when we increased the length of the segments, the *WER* was not greater than the one we obtained for the experiments with the IBM Model 4. We are inclined to think that the language model is better estimated than the distortion model of IBM Model 4.

These results cannot be considered as definitive ones. There is still a technical problem of how to make the search process with the IBM Model 1 using stack-based decoders, that we have not already solved. Specifically, the IBM Model 1 does not provide any information about the most likely zero fertility words that the stack-based decoder needs for to perform the translation (provisionally, we have taken this information from the IBM model 3 fertility model).

6 Conclusions and future works

In our study of the AMETRA task we have discussed the high complexity of the AMETRA task, identifying the main problems that must be dealt with. Obviously, a lot of work must be done if we want to use statistical methods within the memory-translations framework.

Further efforts have to be made about preprocessing, which seems to be the most important difficult here.

Training with the tool *GIZA++* of a segmented corpus obtained from a previous training with the same tool does not seem to be appropriate, because the alignments from which the segments were obtained already had a certain number of errors. For the same reason the translation quality evaluation with automatic measures like *WER* and *PER* were not free of errors either. We plan to investigate groups of words-based translation models in order to eliminate the need of segmenting the corpus.

[5] Search error rate is not given because this feature is not already incorporated to the translator

AMETRA	**WER**	**PER**	**# of segments**
t1	21.2	20.5	711
t2	56.9	56.9	110
t3	51.4	48.5	72
t4	67.0	59.0	64
t5	69.9	57.6	38
t6	73.6	57.2	35
t7	70.2	54.0	31
t8	74.2	50.0	30
t9	67.9	59.8	25
t10	73.0	61.6	26
t11	76.0	62.8	25
t12	78.5	62.8	14
mean	38.5	34.7	

Table 11: Translation results for AMETRA segment corpus using IBM Model 1

In (Al-Onaizan et al., 1999), a study about how to perform the training of a task similar to the AMETRA task is introduced. It might be interesting to follow the guidelines proposed there. Among them, we highlight the use of three toolkits for the Czech language: a lemmatizer, a morphological analyzer and a *POS* tagger. Lemmatized corpora will be also used In the AMETRA project.

In relation to the *POS* tagger, we propose the use of a categorized language model in order to reduce the huge perplexity that the current trigram language model has.

In order to deal with the task complexity, we are considering the adaptation of stack-based decoders for their use as translation assistants where the prediction of short partial hypotheses is made instead of whole sentence translations.

Acknowledgements

This work has been partially supported by the INTEK project under grant TEC-154-02 and by the Spanish CICYT project under grant TIC 2000-1599-C02-01 .

References

Al-Onaizan, Y., Curin, J., Jahr, M., Knight, K., Lafferty, J. D., Melamed, I. D., Purdy, D., Och, F. J., Smith, N. A., and Yarowsky, D. (1999). Statistical machine translation, final report, JHU workshop. `http://www.clsp.jhu.edu/ws99/projects/mt/final\_report/mt-final-report.%ps`.

Berger, A. L., Brown, P. F., Della Pietra, S. A., Della Pietra, V. J., Gillett, J. R., Kehler, A. S., and Mercer, R. L. (1996). Language translation apparatus and method of using context-based translation models. United States Patent, No. 5510981.

Brown, P. F., Della Pietra, S. A., Della Pietra, V. J., and Mercer, R. L. (1993). The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

Germann, U., Jahr, M., Knight, K., Marcu, D., and Yamada, K. (2001). Fast decoding and optimal decoding for machine translation. In *Proc. of the 39th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 228–235, Toulouse, France.

Jelinek, F. (1969). A fast sequential decoding algorithm using a stack. *IBM Journal of Research and Development*, 13:675–685.

Ortiz, D., García-Varea, I., and Casacuberta, F. (2003). An empirical comparison of stack-based decoding algorithms for statistical machine translation. In *New Advance in Computer Vision*, Lecture Notes in Computer Science. Springer-Verlag. 1st Iberian Conference on Pattern Recongnition and Image Analysis - IbPRIA2003- Mallorca. Spain. June.

Wahlster, W., editor (2000). *Verbmobil: Foundations of speech-to-speech translations*. Springer Verlag, Berlin, Germany.