The GREYC Translation Memory for the IWSLT 2009 Evaluation Campaign: one step beyond translation memory
Yves Lepage, Adrien Lardilleux, Julien Gosme

To cite this version:
Yves Lepage, Adrien Lardilleux, Julien Gosme. The GREYC Translation Memory for the IWSLT 2009 Evaluation Campaign: one step beyond translation memory. The 6th International Workshop on Spoken Language Translation (IWSLT09), Dec 2009, Tokyo, Japan. pp.45-49, 2009. <hal-00447076>

HAL Id: hal-00447076
https://hal.archives-ouvertes.fr/hal-00447076
Submitted on 14 Jan 2010
The GREYC Translation Memory for the IWSLT 2009 Evaluation Campaign: one step beyond translation memory

Yves Lepage, Adrien Lardilleux, Julien Gosme
GREYC, University of Caen Basse-Normandie, France
firstname.lastname@info.unicaen.fr

Abstract
This year’s GREYC translation system is an improved translation memory that was designed from scratch to experiment with an approach whose goal is just to improve over the output of a standard translation memory by making heavy use of sub-sentential alignments in a restricted case of translation by analogy. The tracks the system participated in are all BTEC tracks: Arabic to English, Chinese to English, and Turkish to English.

1. Introduction
This paper gives a sketch of the GREYC translation system that participated in the IWSLT 2009 evaluation campaign.

The system participated in all read speech tasks, i.e., the three BTEC tasks with the following source languages: Arabic, Chinese, Turkish, and English as the unique target language.

The following section, Section 2, gives an overview of the preprocessing and the postprocessing that were applied to the data delivered by the organizers with a stress on the different units of processing that were used for each different language. Section 3 describes the different tools used, i.e., the morphological synthetizer for the translation of unknown words, the aligners to produce translation tables, and the translation method. Section 4 describes an experiment to determine a threshold on translation probabilities used to filter out the translation table. Section 5 gives the scores obtained on the test set.

2. Pre- and postprocessing

2.1. Preprocessing: case and punctuation
The data delivered by the organizers of the campaign were consistent between the training set, the devset and the test set on the level of typography. All texts, except Chinese were normal texts in their respective languages. This means, that English and Turkish had capitals and punctuation at the right place. Arabic is segmented in hyperwords1 and without short vowels as in standard texts. Chinese however was delivered segmented into words.

For Turkish as well as for English, we lowercased all texts and tokenized punctuation into words by separating them with blanks. For Arabic, the only preprocessing was to isolate punctuation. To perform all these tasks, we simply unified the three Perl scripts delivered with the data.

2.2. Encoding
All tools designed at GREYC, from the aligner to translation engines or translation tool, are Unicode-compliant. The data delivered during the campaign are encoded in Unicode. SMT practitioners usually apply Buckwalter analyzer to process Arabic. This analyzer delivers a transcription of Arabic that uses ASCII encoding. On the contrary to this approach, we applied our tools directly on the Unicode Arabic texts.

2.3. Unit of processing
Our translation tool, in conformity with what we did in previous years, is capable of processing texts in different unit levels: characters or words. The level of processing that delivers the best results on the devsets was determined after testing for Chinese.

The processing of texts in Turkish, Arabic and English was performed in the units that correspond to graphical words: a series of characters separated by blanks, which means hyperwords for Arabic.

As for Chinese, however, experiments showed that this year’s translation tool performed better on the level of characters than with the segmentation delivered by the organizers. In a setting similar to that of our primary run, the translation of devset1 and devset2 in characters improved the translation quality by 1.43 BLEU points in comparison with the translation in words.

Table 1 summarizes all the above information about preprocessing and unit of processing for the three different tasks the system participated in.
Table 1: Type of pre-processing and processing unit used in the GREYC translation tool. ‘nr’ means not relevant.

<table>
<thead>
<tr>
<th>Language</th>
<th>Lower-case</th>
<th>Isolated punctuation</th>
<th>Processing unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arabic</td>
<td>yes</td>
<td>yes</td>
<td>hyperwords</td>
</tr>
<tr>
<td>Chinese</td>
<td>nr</td>
<td>yes</td>
<td>characters</td>
</tr>
<tr>
<td>Turkish</td>
<td>yes</td>
<td>yes</td>
<td>words</td>
</tr>
<tr>
<td>English</td>
<td>yes</td>
<td>yes</td>
<td>words</td>
</tr>
</tbody>
</table>

2.4. Postprocessing

The three tracks in which the GREYC translation tool participated this year all had English as a target language. We applied Moses’ recaser and detokeniser to glue punctuation back on the outputs of our system to obtain standard English text.

3. The tools used

3.1. Morphological synthesizer

We found 189 unknown hyperwords in Arabic and 134 unknown words in Turkish. These unknown (hyper)words correspond to words that were not found in the translation tables.

Following the proposal in [1] and [2] in the general case of machine translation, and [3] for terminology, we conducted experiments to translate unknown (hyper)words in these two languages by analogy. On the contrary to the approach taken in the papers cited above, the technique we used consisted in producing all possible new word pairs \((a, \hat{a})\) in the source and in the target languages from all possible triples of word pairs \((b \leftrightarrow \hat{b}), (c \leftrightarrow \hat{c}), (d \leftrightarrow \hat{d})\) using an analogy solver written in Python:

\[
\begin{align*}
    x : b & :: c : d \Rightarrow x = a \\
    \hat{x} : \hat{b} & :: \hat{c} : \hat{d} \Rightarrow \hat{x} = \hat{a}
\end{align*}
\]

These analogical equations were solved with the character as the process unit. The (hyper)word-to-(hyper)word translations \((b \leftrightarrow \hat{b}), \ldots\) were filtered from the overall phrase alignments produced from the training data plus the development sets. In addition, ill-formed new (hyper)words are discarded by checking the presence of the n-grams they consist in in the corpus. The n-grams were trigrams for Arabic, Turkish and English, and bigrams for Chinese. The unknown (hyper)words \(a\) are looked for among the produced new (hyper)words. The process being applied in parallel on the source and target languages, their translations \(\hat{a}\) are obtained ‘for free.’

Table 2 gives the number of new words obtained for each language as well as the number of translated unknown words.

The usefulness of this work is however questionable for automatic measures, as the improvement obtained in BLEU scores for the translation of devset1 and devset2 in all three languages were not significant.

3.2. Translation tables

In the same way as standard phrase-based SMT systems (Pharaoh, Moses or Joshua) need translation tables, the tool we designed specifically for this year campaign also requires translation tables obtained by alignment. In this tool, alignments are used to feed the first part of the fundamental operation at work in the translation tool, in collaboration with a translation memory.

We used two different tools to produce translation tables according to the languages: GIZA++ [4] and anyalign [5]. Preliminary experiments performed with both tools for the three language pairs showed better results with GIZA++ for Arabic and Turkish and with anyalign for Chinese. All these experiments were performed on devset1 and devset2 as they were common to the three tasks.

3.3. Translation tool

3.3.1. “Pure” translation by analogy

The principle of translating by analogy [6] is as follows. To translate a new sentence, \(A\), the engine basically solves all possible analogue equations of the type:

\[
A : x :: C : D
\]

where \(C\) and \(D\) are two source text pieces from the training data. If the solution of the equation \(x = B\) belongs to the training data, then its translation \(\hat{B}\) is known and the analogue equation:

\[
y : \hat{B} :: \hat{C} : \hat{D}
\]

can be built and possibly solved in the target language. The principle states that any solution \(y = \hat{A}\) to this equation is a possible translation of \(A\). The analogy solver is non-deterministic and yields all the solutions in such cases.
In addition to all training sentences and all development sentences, last year’s engine [7] leveraged on the use of alignments produced from these data by an aligner.

3.3.2. The principle of translation memory

The principle of memory translation is well-known: it consists in using a set of pre-translated text to help a human translator in his/her task.

In the previous evaluation campaign, the engine designed already used this principle as a back-off strategy: when no solution at all could be found by analogy, the engine output the translation of the source sentence closest to the input sentence. However, a noticeable difference with a plain translation memory was that the set of source sentences used during back-off comprised all new text pieces generated as a by-product during the failed translation process. As for an example, suppose the training set with its alignements were

3.3.3. One step beyond translation memory

The translation tool used in this year departs from the ones used in previous years’ campaign [8, 9, 7] by modifying the approach taken in last years’ campaigns.

In last years’ engines, the translation memory behavior was a back-off strategy. This year, the translation memory strategy was made the very starting point of the overall translation process.

Starting from an input sentence \( A \) the tool first adopts a translation memory behavior: it looks for pairs of sentences \( B_0 \leftrightarrow \hat{B}_0 \) in the source and the target languages such that \( B_0 \) is close to \( A \). The proximity criterion is a normalized edit distance with a unit processing of (hyper)words (except for Chinese, where the processing unit is character). The sentences \( B_0 \) are chosen from the training and development data. They are thus guaranteed to be of the same nature as \( A \), i.e., they are sentences. Experiments have shown that the best results in BLEU scores are obtained with a set of \( B_0 \)’s consisting in the set of sentences closest to \( A \), that is those sentences at a minimal distance to \( A \) (there may exist a plurality of such sentences).

The goal of the subsequent process will be to alter \( \hat{B}_0 \) in such a way that it becomes closer to \( A \). For that purpose, a series of transformations are applied on \( \hat{B}_0 \), that are guided by possible transformations to be found in the set of alignments. Such possible transformations are instantiated by a pair of source text pieces \((C, D)\). They are constrained by imposing that \( D \) be a substring of \( B_0 \) in the source and the target languages (i.e., \( D \) is also a substring of \( \hat{B}_0 \)). In addition, so as to guarantee that the transformation on \( \hat{B}_0 \) will deliver a piece of text closer to \( A \), it is imposed that \( C \) be also included in \( A \). All this leads to the following analogical equation system to be solved:

\[
x : B_0 :: C : D \quad \text{with } D \subset B_0 \text{ and } C \subset A
\]

\[
\hat{x} : \hat{B}_0 :: \hat{C} : \hat{D} \quad \text{with } \hat{D} \subset \hat{B}_0
\]

Let us call \((B_1, \hat{B}_1)\) a solution of the previous system of equations. The same process is recursively applied to \((B_1, \hat{B}_1)\) with \( A \) to guide the process. At each step of the recursive process, all possible pairs of alignments \((C, D)\) are tried. Also at each step \( n \) of the process, the distance between \( B_n \) and \( A \) is estimated. The process is forced to converge by imposing \( d(A, B_{n+1}) < d(A, B_n) \).

The philosophy behind the overall process is to apply transformations that are basically substitutions of \( D \) by \( C \) in \( B_n \)’s. Experiments have shown that choosing \( D \) and \( C \) as contiguous substrings of \( B \) and \( A \) respectively ensure a faster convergence. \( D \) may be the empty string: in this way, deletions of words or sequences of words from \( B_n \) are allowed. By opposition, \( C \) is not allowed to be the empty string, i.e., insertions of words are not allowed into the target sentence when transformed. The reason for this restriction is that the place of insertion in the target sentence cannot be determined for sure.

Among all sentences obtained at different levels of recursion but with the same minimal distance to \( A \), the ones with the lesser recursive level and the highest frequency are chosen.

4. Filtering out alignments by probabilities

At the beginning of the process, the training and development sets are used to select the sentence closest to the sentence to be translated. At each recursion step, pairs of alignments are chosen that meet the constraints explained above. The quality of the translation of these alignments influences the quality of the translation hypothesis. We thus resort to translation probabilities generated during alignment in two places.

Firstly, to ensure a better individual translation quality, at each step of the recursion process, when a phrase gets several possible translations, the one with the best probabilities is the only one to be processed.

Secondly, to ensure a better overall translation quality, a threshold on the translation probabilities of the alignments has been determined experimentally to filter out any alignment with translation probabilities lower than this threshold.
To determine this threshold, a series of experiments has been performed. It consisted in running the system for all possible values of the threshold and measuring BLEU scores obtained in each of the three different language pairs. In those experiments, the training data were the training data delivered as such and the scores where computed on the two development sets devset1 and devset2. Figure 1 shows the evolution of the BLEU scores for the three different languages. For all three languages an increase in BLEU is observed when the threshold increases and a clear maximal value is reached for all three cases. These maximal values were used for the submissions as primary runs.

5. Results

5.1. Comparison with a baseline

As the translation tool has been designed to be one step only away from a translation memory, the baseline of such a tool is a pure translation memory. In an experiment that used the training set to translate the development sets devset1 and devset2, we measured the increase in BLEU. The results are shown in Table 3.

The increase is most significant in Arabic, less significant but reasonable in Turkish, and not so much convincing in Chinese.

5.2. Primary submission

We submitted one primary run for each three tasks. The training set and all development sets are used to find the sentence closest to the sentence to translate and to produce the alignments that are exploited to find possible transformations.

In Table 4, the column ‘# of unknown words’ in the test set has to be understood in the unit of processing for each different source language: hyperwords for Arabic, characters for Chinese and words for Turkish.

Table 5 gives the results in all different measures as delivered by the organizers in the two measuring conditions: without case and punctuation left, and without case nor punctuation.

6. Conclusion

This paper has given a sketch of the GREYC translation memory system that participated in the IWSLT 2009 evaluation campaign in all classical BTEC tasks for the three translation directions: Arabic-to-English, Chinese-to-English and Turkish-to-English. The system intends to make one step away beyond the principle of translation memory by making use of the principle of translation by analogy.

Starting from a sentence that is close to the sentence to be translated, the system applies a series of transformations, extracted as pairs of text pieces from alignments obtained by a standard alignment tool. These pairs of text pieces are constrained to represent a sensible transformation to apply on the close sentence to make it still closer to the sentence to translate, step-by-step.

The principle of corresponding proportional analogies between two languages allows the system to build a possible translation hypothesis in the target language along with the transformation process in the source language.

7. References


Figure 1: Determination of the threshold for translation probabilities. The threshold is expressed as the geometric mean of the two conditional probabilities: the source knowing the target and the target knowing the source. The BLEU scores grow before reaching a maximum at different values and fall down slightly afterwards. The experimental values retained are thus: 0.25 for Arabic, 0.5 for Chinese, and 0.4 for Turkish.

Table 5: Scores according to all different metrics obtained for the primary run of the improved translation memory, in the two measurement settings: with case and punctuation left and without case and punctuation. The columns ‘f1’ (harmonic mean of precision and recall), ‘prec’ (precision) and ‘recl’ (recall) pertain to meteor.

<table>
<thead>
<tr>
<th>task</th>
<th>case/punc</th>
<th>bleu</th>
<th>meteor</th>
<th>f1</th>
<th>prec</th>
<th>recl</th>
<th>wer</th>
<th>per</th>
<th>ter</th>
<th>gtm</th>
<th>nist</th>
</tr>
</thead>
<tbody>
<tr>
<td>BTEC_AE</td>
<td>case+punc</td>
<td>0.329</td>
<td>0.617</td>
<td>0.686</td>
<td>0.734</td>
<td>0.644</td>
<td>0.512</td>
<td>0.453</td>
<td>43.262</td>
<td>0.661</td>
<td>5.654</td>
</tr>
<tr>
<td></td>
<td>no_case+no_punc</td>
<td>0.307</td>
<td>0.566</td>
<td>0.633</td>
<td>0.688</td>
<td>0.587</td>
<td>0.587</td>
<td>0.510</td>
<td>48.836</td>
<td>0.623</td>
<td>5.536</td>
</tr>
<tr>
<td>BTEC_CE</td>
<td>case+punc</td>
<td>0.280</td>
<td>0.554</td>
<td>0.613</td>
<td>0.637</td>
<td>0.590</td>
<td>0.592</td>
<td>0.532</td>
<td>51.609</td>
<td>0.596</td>
<td>5.657</td>
</tr>
<tr>
<td></td>
<td>no_case+no_punc</td>
<td>0.277</td>
<td>0.510</td>
<td>0.564</td>
<td>0.591</td>
<td>0.539</td>
<td>0.655</td>
<td>0.579</td>
<td>57.212</td>
<td>0.565</td>
<td>5.927</td>
</tr>
<tr>
<td>BTEC_TE</td>
<td>case+punc</td>
<td>0.355</td>
<td>0.648</td>
<td>0.708</td>
<td>0.745</td>
<td>0.674</td>
<td>0.509</td>
<td>0.437</td>
<td>41.633</td>
<td>0.678</td>
<td>6.347</td>
</tr>
<tr>
<td></td>
<td>no_case+no_punc</td>
<td>0.346</td>
<td>0.600</td>
<td>0.658</td>
<td>0.704</td>
<td>0.617</td>
<td>0.570</td>
<td>0.484</td>
<td>47.052</td>
<td>0.644</td>
<td>6.425</td>
</tr>
</tbody>
</table>


