Intelligent Information Systems 9999
ISBN 666-666-666, pages 1–12

Building high quality translation memories acquired from monolingual corpora

Rafał Jaworski[1] and Krzysztof Jassem[1]

Adam Mickiewicz University, Poznań, Poland

Abstract

The paper presents two ideas for overcoming the problem of translation memory data sparseness. The first is specialization in a narrow domain. The second idea is a novel method of preparing a specialized translation memory for a given purpose. It is based on the assumption that the most useful sentences that might appear in a domain-restricted translation memory are those which occur most frequently in texts from this domain. A clustering algorithm classifies sentences from a monolingual corpus into clusters of similar sentences and selects one representative for each cluster. The translation for each representative is then produced manually by human specialists. The database prepared in such a manner forms a high-quality specialized translation memory, covering a wide range of domain-specific sentences intended for translation.

Keywords: translation memory, EBMT, CAT, clustering

1 Introduction

As fully automatic translation often falls short of expectations, people turn to other methods of using computers in the translation process. One of such methods is Computer-Aided Translation (CAT).

CAT systems use sets of previously translated sentences, called the translation memories. For a given sentence, a CAT system searches for a similar sentence in the translation memory. If such a sentence is found, its translation is used to produce the output sentence. This output sentence is then used as a suggestion for translation. A human translator carries out the post-editing.

It is crucial for a CAT system that the translation memory contain translations of frequently used sentences. With such a translation memory, a CAT system generates good suggestions for translations, reducing the amount of human work on post-editing.

Translation memories in CAT systems are most commonly created dynamically on the basis of translations provided by user. For some applications, however, it is required from a CAT system to deliver the initial high quality domain-specific translation memory. This is the case for the LEX project, described in Section 3, where the ordering party, Wolters Kluwer Polska, requires the system to contain the Polish-English translation memory of sentences used in legal contracts.

So far, translation memories have been usually acquired automatically from bilingual corpora available on the Internet (see Steinberger *et al.* 2006, Eur-Lex 1998-2010). Initially, we followed this idea, incorporating among other sources, the well-aligned corpus of legal texts, JRC (Steinberger *et al.*, 2006). The evaluation experiments showed that the resulting translation memory was of little help for human translators.
We have identified the following reasons for this state:

- Aligning algorithm used to obtain translation memories from bilingual corpora leave a margin of wrong pairings (Lipski, 2007)
- Translations found in the bilingual corpora are not perfect
- Bilingual data for a specific domain are sparse
- To overcome sparseness it is needed to include large numbers of translation units. This has negative influence on search complexity

The paper presents two ideas for overcoming the above problems. The first is specialization in a narrow domain (inspired by Gough and Way 2003). By limiting the range of texts for translation we restrict both vocabulary (to domain-specific terms) and grammar (to most common constructions). In our project the use of the translation memory has been limited to legal texts.
The second idea is a novel method of preparing a specialized translation memory on the basis of monolingual corpora, which are much easier to collect for a specialized domain. The idea is based on the assumption that the most useful sentences that might appear in a domain-restricted translation memory are those, which occur most frequently in texts from this domain. A clustering algorithm classifies sentences from a monolingual corpus into clusters of similar sentences and selects one representative for each cluster. The translation for each representative is then produced manually by human specialists. The database prepared in such a manner forms a high-quality specialized translation memory, covering a wide range of domain-specific sentences intended for translation.
As the functioning of a CAT system depends strongly on the Example-Based Machine Translation (EBMT) mechanism it uses, the idea of EBMT is described in general in Section 2. The LEX project is described briefly in Section 3. Section 4 presents the EBMT implementation used in LEX. Section 5 describes the clustering algorithm. The evaluation of the algorithm is presented in Section 6.

2 Example-Based Machine Translation

The idea of basing automatic translation of a sentence on a previously translated similar example is known as Example-Based Machine Translation (EBMT). Numerous EBMT systems have been developed since 1984, when Makoto Nagao first suggested the idea (Nagao, 1984). However, over the past few years researchers have been turning away from EBMT as another translation paradigm – Statistical Machine Translation (SMT) has taken over the field of machine translation based on translation memories (Forcada and Way, 2009).
One of the reasons of this process is a certain drawback of EBMT: An EBMT system is likely to produce a high-quality translation provided that an appropriate

example is found in the translation memory (Smith and Clark, 2009). However, in most cases it is impossible to find such an example, due to data-sparseness constraints (Smith and Clark, 2009). This greatly limits the usability of an EBMT system. But if it could be overcome, the resulting machine translation system would be a powerful tool.

3 The LEX Project

LEX is a project aimed at developing a CAT tool for translating legal texts. It involves creating legal glossaries and translation memories as well as improving CAT techniques.
The creation of glossaries and translation memories involves the participation of professional translators. First, a legal translation memory is obtained, either downloaded directly from the Internet, or compiled from bilingual texts (using automatic alignment mechanisms). Translation memories serve as a source for bilingual dictionaries of legal phrases, which are extracted automatically and verified by professional linguists.
Polish-English translation memories collected so far include: the Eur-Lex translation memory (Eur-Lex, 1998-2010), the Polish constitution and acts on: higher education, copyrights, employment and company law. The total number of translation units exceeds 4.5 million.
Surprisingly, experiments have shown that this large translation memory itself is of little help for translation of legal contracts.

4 EBMT implementation

4.1 General information

The LEX project uses an EBMT module, called NeLex. According to the definitions in Carl *et al.* (2003) and Turcato and Popowich (2001), NeLex implements the idea of "pure" EBMT: NeLex uses translation memory. For each translated sentence an example is searched for in the memory before the transfer process is initiated. Like the EBMT system designed at the Chinese Academy of Science (Hongxu *et al.*, 2004), NeLex consists of two basic modules: Example Matcher and Transferer. The former is responsible for finding in a translation memory an example best suited for the input sentence. The latter modifies the example target sentence so that it can be returned as the translation of the input sentence.

4.2 Word substitution

NeLex's Transferer module performs a sequence of operations to produce the translation of the input sentence. One of them is word substitution. The mechanism of this substitution is illustrated by the following example:

INPUT SENTENCE (in Polish): "Uwzględniając Traktat ustanawiający Parlament Europejski".
(in English: Having regard to the Treaty establishing the European Parliament).

Example from the translation memory:
SOURCE SENTENCE (in Polish): "Uwzględniając Traktat ustanawiający Wspolnotę Europejską".
TARGET SENTENCE (in English): "Having regard to the Treaty establishing the European Community."

The translation result is:
"Having regard to the Treaty establishing the European Parliament".

Note that the word Community in the example was substituted by the word Parliament from the input sentence. This simple-looking operation requires a correct word-alignment between sentence pairs. More information on word-alignment in NeLex may be found in Jaworski (2009).

4.3 Named Entity Recognition and Translation

The NeLex system is also capable of substituting larger parts of sentences - Named Entities. Following Graliński *et al.* (2009), we define a Named Entity as a continuous fragment of text referring to information units such as persons, geographical locations, names of organizations or locations, dates, percentages or amounts of money. Named Entity Recognition plays a key role in the process of Machine Translation, as stated in Graliński *et al.* (2009). Usually, Named Entities carry the most important information in the sentence. Experience shows (as stated in Vilar *et al.* 2006) that Named Entities are prone to translation errors. Hence, correct handling of Named Entities during the process of translation can considerably improve the translation quality (Graliński *et al.*, 2009). Table 1 lists some Named Entity types handled by NeLex.

Table 1: Common Named Entity types.

Type	Description	Example(s)
JOLFull	Reference to the Journal of Laws	Journal of Laws of 2004/04/06
Company	Name of a corporation	ACME Ltd.
E-mail	An e-mail address	login@example.com
Paragraph	Reference to a paragraph	§34 sec. 4 item g)
Number	A real number	315.871

In NeLex, Named Entities are managed by a substitution mechanism similar to word substitution. Recognition and translation of Named Entities is executed by a dedicated module (see Jaworski 2009), which uses semi-supervised learned rules. The rules are written in a formalism called NERT (full specification of the formalism can be found in Graliński *et al.* 2009). The NeLex formalism applies only a part of the NERT formalism: it disregards linguistic knowledge about words. A simple example of a NERT rule is presented below:

```
Direction: Polish to English
Match: <[A-Z]\w+> Sp\. z o\.o\.
Action: replace(\1 Ltd.)
```

The above rule applies to translation from Polish to English. It serves to substitute a Polish name of a company with its appropriate English equivalent.

4.4 Specialization in Legal Texts

The NeLex's key feature is specialization in a dedicated domain of texts (the idea is described in general in Gough and Way 2003), i.e. in legal texts. NeLex deals with the characteristic features of texts in this domain, such as: legal vocabulary and the presence of legal references, such as 'article 23 No. 78, item 483'.
Much of the most common legal vocabulary is contained in NeLex's dictionaries and used during word substitution. Legal references, on the other hand, are treated as a type of Named Entities. Moreover, legal texts are likely to be translated correctly with the use of translation memory, as they contain recurring sentences and phrases, such as: 'with subsequent amendments', 'with regard to the provisions of...', 'this article has been repealed' and many others.

5 The Algorithm for Sentence Clustering

5.1 The Need for the Algorithm

In the first step of creating translation memory for the Lex project we collected a large translation memory (see Section 3). However, usability tests have revealed significant disadvantages of this translation memory.
One disadvantage is a low speed of example search. Although the search mechanism has been optimized, it still requires considerable amount of time to find a good example for a given input sentence (especially for long input sentences) because of the translation memory size. This problem limits the usability. Moreover, when the translation system has been tested on real-life examples, it has become clear that the Eur-Lex translation memory, though vast, does not contain examples of sentences commonly used by lawyers. This is because it only contains the Official Journal of the European Union as well as the treaties, legislation, case-law and legislative proposals (Eur-Lex, 1998-2010). It does not, for instance, contain any contracts, which is the main type of documents translated for lawyers.
A closer look into our Translation Memory revealed that not all translations acquired automatically exhibit high quality.

5.2 Clustering – the Main Idea

Having discovered these problems, we developed a new method of developing translation memory: in the first step a monolingual corpus is selected, containing the types of texts most likely to be translated with the aid of the system. Such a monolingual corpus is obviously easier to obtain than an equivalent bilingual corpus. In the LEX project, we compiled a corpus containing various contract and

pleading templates, as well as court decisions from the Polish legal text database commonly used by lawyers (LEX Prestige system).
The idea of clustering is based on the observation that many sentences in the corpus exhibit certain similarities. Thus, a valuable translation memory can be obtained by producing the translations of the most frequent sentences in the corpus. Indeed, translating all the sentences from the corpus would be time-consuming and labour-intensive, yet for a relatively small list of most typical sentences it is possible to translate them manually.
In the LEX project, a list of approximately 500 most typical Polish sentences used in contracts were extracted by means of the clustering algorithm. The sentences were then translated into English by professional translators.

5.3 The Clustering Algorithm

Clustering algorithms work on sets of objects. They use a measure of distance between these objects in order to divide sets into smaller chunks containing objects which are "close" to each other (in terms of the distance measure). In our case, the set of objects is the monolingual corpus, the objects are sentences. In order to determine the distance between sentences we use two distance measures: "cheap" and "expensive" (the idea of using two distance measures is inspired by McCallum *et al.* 2000). The "cheap" and "expensive" terms refer to complexity of calculations.

Cheap sentence distance measure

The cheap sentence distance measure is expected to work fast. It is based only on the lengths of sentences. The formula for computing the value of the measure is the following:

$$d_c(S_1, S_2) = 2^{-\frac{|C_{S1} - C_{S2}|}{10}}$$

Where:
C_{S1} – the length of the sentence S_1 (in characters)
C_{S2} – length of the sentence S_2 (in characters)

Expensive sentence distance measure

The expensive sentence distance measure is expected to give more accurate assessment of sentence similarity. The first step of computing the distance between sentences S_1 and S_2 in this measure is to recognize Named Entities of the two sentences (using the mechanism described in Section 4.3). Then, so called "penalties" are imposed for discrepancies between the sentences. The penalty values are presented in Table 2. The penalty values have been based on human translators' intuition, e.g. if one sentence contains a Named Entity and the other does not, then the sentences are not likely to be similar. (The values for penalties are bound to be calculated by self-learning techniques in future experiments).

Table 2: Penalty values for discrepancies.

Discrepancy	Penalty value
Lexical correspondence (not identity) of words	0.5
Type correspondence (not identity) of Named Entities	0.5
Word from S_1 missing in S_2	1.0
Word from S_2 missing in S_1	1.0
Named Entity from S_1 missing in S_2	1.5
Named Entity from S_2 missing in S_1	1.5
Inversion of words or Named Entities	0.5
Missing punctuation mark	0.25

Let us define:
p – the sum of penalties imposed on sentences S_1 and S_2
L_{S1} – the number of words and Named Entities in S_1
L_{S2} – the number of words and Named Entities in S_2

$d_e(S_1, S_2) = 1 - \frac{2p}{L_{S1}+L_{S2}}$

The clustering procedure

The detailed clustering algorithm is:

```
IN: Set (S) of strings (representing sentences)
OUT: List (C) of clusters (representing clusters of sentences)

1. Divide set S into clusters using the QT algorithm with the "cheap"
    sentence distance measure (the measure is based only on sentence length).
2. Sort the clusters in the descending order by number of elements
    resulting in the sorted list of clusters, C.
3. For each cluster cL in list C:
(a) Apply the QT algorithm with the "expensive" distance measure
    (based on sentences contents) to cL, resulting  in subclusters.
(b) Sort the subclusters in cL in descending order by number of elements.
(c) Copy the sentences from the cL into C
```

The QT algorithm is described in Heyer *et al.* (1999). The final step is performed by humans. They manually select the most valuable sentence from each clusters. The task is facilitated by a frequency prompt (the most frequent sentences always appear at the beginning of the cluster).

6 Evaluation

The translation memory prepared with the help of the clustering algorithm is expected to improve the quality of suggestions for translations produced by the CAT system, reducing the amount of human work needed to translate a document. This expectation has been verified by the following experiment:
A test set consisting of sentences from 3 contracts was selected. The contracts were selected from a Polish monolingual corpus of legal texts developed in the

LEX project. The total number of sentences in the contracts was **379** and they constituted the test set T_c.
Sentences from T_c were then translated using the CAT system described in Section 4. The EBMT mechanism used in this system was capable of translating a part of sentences from T_c, for which similar sentences were found in the translation memory. Human translators were asked to deliver the final translation for these sentences, having the EBMT translations as prompts.
The idea was to compare the amount of work carried out by translators with and without the specialized translation memory, acquired via the clustering algorithm. Two of the translators took part in Version A of the experiment, while the other two – in Version B. All the translators were working independently.

6.1 Version A of the experiment

In this version, the CAT system used to produce suggestions was trained by the translation memory M that consisted of:

- Bulletin about European Union Legislation
- Polish Penal Code
- Code of Criminal Procedure
- Bilingual corpus developed in the LEX project (not containing the contracts to be translated)
- Act on Copyright
- Act on Education

The total number of translation units in M was: **20301**.
Out of the 379 sentences of T_c, the total of **35** sentences were translated with the help of translation memory M (they constituted the test set T1). Therefore, the recall parameter (the percentage of translated sentences) was **9.2**%.

6.2 Version B of the experiment

In Version B of the experiment, the CAT system was trained with the translation memory M_{500}, which consisted of:

- The translation memory M
- 464 translated units selected with the use of the clustering procedure

The total number of translation units in M_{500} was: **20765**.
In this case, out of 379 sentences of T_c, **118** sentences were translated by the CAT system. It gave a considerably higher recall of **31.1**%.

6.3 Measuring translators' amount of work

After the translation had been performed by humans, the amount of work carried out by them was assessed. This was accomplished with the use of two measures. The first computes amount of work needed to translate a sentence when a suggestion is given. The second is used to assess the effort needed to translate a sentence from scratch, when no suggestion is available.

Effort measure – with suggestion

The measure of effort needed to translate a sentence with a given suggestion is computed by comparing the suggestion to the final human translation. Generally speaking, the more edit operations are needed to be performed on the suggestion in order to produce the final translation, the higher effort measure.
An edit operation is one of the following:

- Deleting a word from the suggestion
- Inserting a word into the suggestion
- Replacing a word in the suggestion

This way of computing the distance between sentences is based on Word Error Rate (WER, described in Hunt 1990), operating on the Levenshtein distance (described in Levenshtein 1965).
For the sake of this experiment, the Lemmatized Word Error Rate measure has been developed. In this measure, before applying the WER, both sentences that are to be compared undergo the following operations:

1. Lemmatize each word of the sentence
2. Remove words shorter than 3 characters from the sentence
3. Remove repetitions of words in a sentence

By lemmatizing the words we assure that the measure does not treat changing the form of a word as significant effort. The same applies to operations on short words. Moreover, the measure does not count the single effort to translate a word more than once. Consider the following example (an actual example from the experiment):

Suggestion: "the object of the partnership's activity"
Translation: "The objects of the Company shall be"

After step 1:
Suggestion: "the object of the partnership activity"
Translation: "the object of the company shall be"

After step 2:
Suggestion: "the object the partnership activity"
Translation: "the object the company shall"

After step 3:
Suggestion: "the object partnership activity"
Translation: "the object company shall"

The computed Lemmatized Word Error Rate value (the effort to produce translation) in this case equals 2 (because of two substitutions).

Effort measure – without suggestion

When the suggestion for translation is not available, a different effort measure is applied. The measure is roughly based on the number of unique words in the source sentence (if the word repeats, the translator only needs to look up its translation once). However, the source sentence undergoes modifications similar to those in the Lemmatized Word Error Rate computation procedure. The operations are:

1. Remove words shorter than 3 characters from the sentence
2. Remove repetitions of words

Lemmatizing is not necessary in this case, as we are only interested in the number of words (lemmatized or not) remaining in the sentence after executing steps 1 and 2.

Consider this example:
Source: "The date of the confirmation of payment by a post office shall be deemed the date of payment"

After step 1 (removed short words "of", "by", "a", "be" and the second "of"):
Source: "The date the confirmation payment post office shall deemed the date payment"

After step 2 (removed repetitions "the" and "payment"):
Source: "The date confirmation payment post office shall deemed date"

In this case, the computed effort value is 9.

6.4 Results and conclusions

Table 3 presents the results of the experiment. Here are explanations of the table entries:

- **Sentences translated** – the number of sentences from T_c that have been translated with the EBMT algorithm and post-edited by human translators
- **Effort for translated sentences** – the computed effort value for translated sentences
- **Total effort** – the computed effort value for both translated sentences and those left untranslated (not found in the translation memory)

In Table 4, the translation memory statistics are presented.

The experiment results, along with comments from the translators, indicate a considerable improvement of quality of the suggestions when using memory M_{500}. The computed average effort per translating one sentence using a suggestion decreased from 6.06 to 2.99. Apart from that, the amount of sentences, for which the translation suggestion was available, was over 3 times higher. Both these factors lead to reducing the total effort needed to translate the sentences from the T_c test set. The total effort value decreased from 3851 to 3366.5. All these improvements

Table 3: Experiment results.

	Version A		Version B	
	Person 1	Person 2	Person 1	Person 2
Sentences translated	35	35	118	118
Effort for translated sentences	210	214	362	343
Average effort per translated sentence (average of Person 1 and 2)	6.0	6.11	3.07	2.91
	6.06		**2.99**	
Total effort (average of Person 1 and 2)	3849	3853	3376	3357
	3851		**3366.5**	

Table 4: Translation memory statistics.

	M (Version A)	M_{500} (Version B)
Total number of units	20301	20765
Size difference to M	0%	+2.3%
Recall score	9.2% (35 of 379)	31.1% (118 of 379)

were achieved thanks to augmenting the translation memory capacity for the CAT system by merely 2.3%.

References

Michael Carl, Andy Way, *et al.* (2003), *Recent Advances in Example-Based Machine Translation*, Kluwer Academic Publisher.

Eur-Lex (1998-2010), Access to European Union law, http://eur-lex.europa.eu/.

Mikel Forcada and Andy Way (2009), Foreword, Proceedings of the 3rd International Workshop on Example-Based Machine Translation.

Nano Gough and Andy Way (2003), Controlled Generation in Example-Based Machine Translation, http://www.mt-archive.info/MTS-2003-Gough.pdf.

Filip Graliński, Krzysztof Jassem, and Michał Marcińczuk (2009), An environment for named entity recognition and translation, *Proceedings of the 13th Annual Conference of the EAMT, Barcelona.*

Laurie Heyer, Semyon Kruglyak, and Shibu Yooseph (1999), Exploring Expression Data: Identification and Analysis of Coexpressed Genes, *Genome Research 9:1106-1115.*

Hou Hongxu, Deng Dan, Zou Gang, Yu Hongkui, Liu Yang, and Xiong Deyi (2004), An EBMT System Based on Word Alignment, http://www.mt-archive.info/IWSLT-2004-Hou.pdf.

Melvyn Hunt (1990), Figures of Merit for Assessing Connected Word Recognisers, *Speech Communication, 9, pages 239-336.*

Rafał Jaworski (2009), Tłumaczenie tekstów prawniczych przez analogie, *Master thesis under the supervision of dr Krzysztof Jassem.*

Vladimir Levenshtein (1965), Binary codes capable of correcting deletions, insertions, and reversals, *Doklady Akademii Nauk SSSR, 163(4):845-848.*

Jarosław Lipski (2007), Urównoleglanie tekstów dwujęzycznych na poziomie zdania, *Master thesis under the supervision of dr Krzysztof Jassem.*

Andrew McCallum, Kamal Nigam, and Lyle H. Ungar (2000), Efficient clustering of highdimensional data sets with application to reference matching, http://www.kamalnigam.com/papers/canopy-kdd00.pdf.

Makoto Nagao (1984), A framework of a mechanical translation between japanese and english by analogy principle, *Proceedings of the International NATO Symposium on Artificial and Human Intelligence, pages 173–180.*

James Smith and Steven Clark (2009), EBMT for SMT: A New EBMT-SMT Hybrid, Proceedings of the 3rd International Workshop on Example-Based Machine Translation.

Ralf Steinberger, Bruno Pouliquen, Anna Widiger, Camelia Ignat, Tomaž Erjavec, Dan Tufiş, and Dániel Varga (2006), The JRC-Acquis: A multilingual aligned parallel corpus with 20+ languages, Proceedings of the 5th International Conference on Language Resources and Evaluation.

Davide Turcato and Fred Popowich (2001), What is Example-Based Machine Translation?, http://www.iai.uni-sb.de/ carl/ebmt-workshop/dt.pdf.

David Vilar, Jia Xu, and Hermann Ney Luis Fernando d'Haro (2006), Error Analysis of Statistical Machine Translation Output, Proceedings of the Language Resources and Evaluation conference.