Expanding Translation Memories: Proposal and Evaluation of Several Methods

Vít Baisa, Josef Bušta, and Aleš Horák

NLP Centre, Faculty of Informatics
Masaryk University
Botanická 68a, Brno, Czech Republic
{xbaisa, xbusta1, hales}@fi.muni.cz

Abstract. Translation memories used in Computer-aided translation (CAT) systems are the highest-quality resources of parallel texts since they are carefully prepared and checked by professional human translators. On the other hand, they are quite small when compared with other parallel data sources. In this paper, we propose several methods for expanding translation memories using both language-independent and language-specific, linguistically motivated approaches with regard to preserving their high translational accuracy. We first briefly describe the methods and then we provide a detailed description and preliminary evaluation for two of them.

Key words: translation memory, computer-aided translation, expanding translation memories

1 Introduction

Translation memory is a set of *translation pairs* containing segments from text documents which were previously manually translated by human translators. These segments (which might refer to sentences, paragraphs, list items, headings, titles etc.) can be then reused within the CAT[1] process and save both time and effort of human translators.

Since translation memories are built manually by human expert translators, they are a) relatively small in comparison with other parallel resources, e.g. *OPUS* [1], *Europarl* [2], or *JRC-Acquis* [3], and b) usually are not available freely as being a property of a professional translation companies, despite some exceptions as e.g. *MyMemory* [4].

But at the same time (as for many other NLP[2] related tasks) this holds good: the bigger is a translation memory, the better is the CAT process. Where better here means faster, of higher quality etc.

The purpose of this paper is thus to present several methods for expanding translation memories using available resources and tools. These methods

[1] Computer-aided translation
[2] Natural language processing

Aleš Horák, Pavel Rychlý (Eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2013, pp. 71–78, 2013.

exploit different amount of linguistic knowledge: from purely statistical and *n*-gram based to syntactic-semantic processing of the input translation memories.

The first method with its variant is described in detail in this paper and its preliminary evaluation is given. We are interested mainly in English-Czech and Czech-English translation pairs.

2 Related Work

Translation memories (TM) are understudied resources in the realm of NLP. They are often presented [5] within a closely related field: *example-based machine translation* (*EBMT*) which uses a similar approach as CAT systems do – reusing samples of previously translated texts.

The TM related papers mainly focus on algorithms for searching, matching and suggesting segments within CAT systems [6] but not much work was devoted to the problem of expanding translation memories.

In [7], the authors have attempted to build translation memories from Web since they found that human translators in Canada use Google search results even more often than specialized translation memories. That is why the research team at the National Research Council of Canada developed a system called *WeBiText* for extracting possible segments and their translations from bilingual webpages. They state an important notice: it is always better to provide translators with a list of possible translations and let them find the correct one than to have nothing prepared. In other words, it is easier and faster for the translators to look up a good translation than to make up their own translation from scratch. Also, it is very important that the correct translation must be between the first 10 or 20 items in the suggested list.

WeBiText system successively tested two approaches: an on-demand version which took a user query (expression) and then asked a search engine for all results. For these results it then tried to find links to their mutation in a target language. This approach was very slow so they resorted to another approach: an off-line version with precompiled results.

In the study [8], the authors exploited two methods of segmentation of translation memories. Their approach is probably the most similar to our subsegment combination method presented below. The main difference is that we use statistical methods of the phrase-based machine translation (PBMT) approach [9] for extraction of new translation pairs of segments.

[10] describes a method of subsegmenting of translation memories which deals with the principles of EBMT. The authors of this study created an on-line system TransSearch [11] for searching possible translation candidates within all subsegments in already translated texts. These subsegments are linguistically motivated – they use a text-chunker to extract phrases from the Hansard corpus, a text corpus containing the Canadian parliamentary debates from 1803 to the present time.

3 Expanding Translation Memories

In following text, we describe four methods which deal with TM enlarging in this way: for a given translation memory TM and a document D to be translated, take TM and try to enlarge it for the purpose of translation of the document D. For this, either various additional resources (parallel corpora) and tools for generalising available data (morphological, syntactic and semantic analysis) are used extensively.

3.1 Method A – Subsegment Combination

The first method uses a parallel corpus (OPUS [1]) and trains a translation model M^t with the *GIZA++* tool [12,13]. Then it takes TM and extracts all consistent phrases from all aligned segments in TM using appropriate word matrices (see Figure 1) yielding TM_{sub}, a translation memory of subsegments.

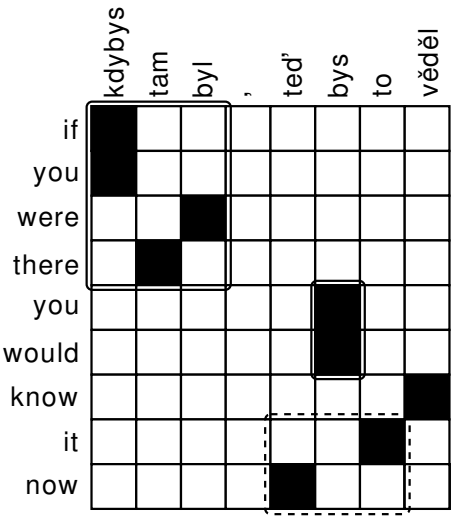

Fig. 1: Word matrix for two aligned sentences / segments.

Word matrices are built directly from the M^t translation model. M^t defines conditional translation probabilities between all pairs of source and target words from the parallel corpus (OPUS). E.g. $p(\text{pes}|\text{dog}) = 0.79$ means that there is 79% probability that the source word *dog* (English) will be translated into the target word *pes* (Czech).

If a pair of two words has sufficiently high probability (higher than a threshold) then in the corresponding word matrix there is a black cell for the pair (see Figure 1). The probability threshold was set experimentally to 0.01 and will be experimentally tuned in the future.

All consistent phrases are then extracted from the word matrices. Consistent phrase is pair of two ranges of words: words from i to j from a source sentence s (s_{ij}) and words from k to l from the corresponding target sentence t (t_{kl}). We

regard s_{ij} and t_{kl} to be consistent when all translations (represented as black cells) of words between the positions i and j are inside the interval from k to l in the target sentence.

Examples of two consistent phrases are displayed in Figure 1 and are outlined with solid line. The third dashed outlined phrase is an inconsistent phrase since it violates the condition. The extracted consistent phrases then form the TM_{sub} translation memory of subsegments.

The memory of subsegments then serves as a basis for building TM_{exp}, the expanded translation memory by combining subsegments that partially match with (sub)segments from the translated document. Each new segment in TM_{exp} must be created as a result of one of the following operations:

a) **join** – new segments are built by concatenating two other segments from TM and TM_{sub}
b) **substitute** – new segments can be created by replacing a part of one segment with a whole of another (sub)segment from TM and TM_{sub}.

An evaluation of the subsegment combination method is presented in section 4.

3.2 Method B – Subsegment Lexicalization

This method is a generalisation of the previous method using linguistic preprocessing: all segments are tokenized and lemmatized and the searching and matching operations work on lemmata. The two corresponding combination operations are now:

a) **ljoin** – concatenation of two different segments from TM and TM_{sub} but this time on lemmata; when concatenating into new resulting segments, appropriate word form (case, gender and number) is generated in the target language
b) **lsubstitute** – substitution of a part of target segment with another segment but again using lemmata and generating proper word forms with correct case, gender and number in the target language

With this method we expect increasing the recall (coverage) but at the same time not decreasing the translation accuracy of original segments from TM. So it is partially rule-based method.

3.3 Method C – Machine Translation of Subsegments

The process of this method follows the previous methods A and B with two other combining operations:

a) **substtran** – new segment is created by translating its part by a freely available machine translation systems

b) **lsubsttran** – combination of **substran** and **lsubstitute** operations – the translation is done for segment parts (phrases) in basic form and the translation result is then transformed into correct word forms.
For example, the Czech phrase *modré knížce* ("to [the] blue book") has its basic phrase form *modrá knížka* ("blue book") which is different from phrase *modrý knížka* where all words from the phrase are in the base form: gender agreement must hold for base forms of phrases.

Example: Let us have a sentence s_s in TM: *Návod na použití desinfekčního přípravku najdete na konci této brožury* together with its proper translation t_t: *You can find instructions for use of disinfectant at the end of this brochure*, and a sentence s_d in D to be translated: *Návod na použití kartáče na vlasy najdete na konci této brožury*. Given that subsegment *kartáče na vlasy* is not in previously built TM_{sub} we need to get translation of it. Google Translate gives us *hairbrush* as translation of the base form. So the only thing to do is to identify the subsegment, put it into its base form, translate it with some of MT systems and substitute the appropriate part of target segment to be able to translate the whole sentence s_s.

3.4 Collocation-Based Filtering to Expanding TM

The previous methods often generate too many candidates for the TM_{exp} expanded translation memory. The CAT systems offer the possibility to sort the translation memory *matches* expressed as a percentage of the correspondence between the segment from TM and segment from the translated document. We thus use a value of *collocability* of the target segment phrase as a base for this percentage. The collocability is a number showing how common the phrase and its parts are in the language. In this way, we prefer those translations that correspond to frequently used phrases.

4 Evaluation

Authors of [10] reported 28% coverage with precision 37% for 100 test sentences. For evaluation purposes, we have used a different test data so it is not straightforward to compare the two results.

As test data we have used a sample of translation memory TM^s and an example document D^s provided by one of the biggest Czech translation company.

The presented results have been obtained directly from the pre-translation analysis of the MemoQ CAT system.[3] The numbers express how many segments from the document D^s can be translated automatically by MemoQ. The automatic translation is done on the segment level and even on lower levels: on levels of subsegments. Various matches on lines in the table correspond to these

[3] `http://kilgray.com/products/memoq`

Table 1: TM analysis for the first phase of the subsegment combination method A, without the *join* operation

	TM^s				TM_{sub}				TM^s+TM_{sub}			
Match	**Seg**	**wrds**	**chars**	**%**	**Seg**	**wrds**	**chars**	**%**	**Seg**	**wrds**	**chars**	**%**
100%	23	128	813	0.4	2	5	23	0.01	25	133	836	0.38
95–99%	45	185	1 130	0.5	296	363	1 924	1.03	305	480	2 620	1.37
85–94%	4	21	155	0.1	20	54	337	0.15	24	75	492	0.21
75–84%	42	208	1 305	0.6	85	237	1 474	0.67	102	358	2 258	1.02
50–74%	462	1 689	10 293	4.8	772	4 031	24 826	11.47	784	4 449	27 370	12.66
any match	576	2 231	13 696	6.4	1 175	4 690	28 584	**13.33**	1 240	5 495	33 576	**15.64**

Table 2: TM analysis for the subsegment combination method A including the **join** operation

	TM^s				$TM_{subjoin}$				$TM^s+TM_{subjoin}$			
Match	**Seg**	**wrds**	**chars**	**%**	**Seg**	**wrds**	**chars**	**%**	**Seg**	**wrds**	**chars**	**%**
100%	23	128	813	0.4	2	5	23	0.01	25	133	836	0.38
95%–99%	45	185	1 130	0.5	296	363	1 924	1.03	305	480	2 620	1.37
85%–94%	4	21	155	0.1	20	54	337	0.15	24	75	492	0.21
75%–84%	42	208	1 305	0.6	88	255	1 565	0.73	102	358	2 258	1.02
50%–74%	462	1 689	10 293	4.8	787	4 256	26 136	12.11	798	4 629	28 423	13.17
any match	576	2 231	13 696	6.4	1 193	4 933	29 985	**14.03**	1 254	5 675	34 632	**16.15**

sublevels – 100% match corresponds to the situation when a whole segment from D^s can be translated using a segment from the available TM. Translations of shorter parts of the segment are then matches lower than 100%.

The analysis results provided by CAT systems are usually used for estimating the amount of work needed for the (human) translation of a given document and subsequently for estimating the price of the translation work. The higher number of segments which can be translated automatically, the lower is the price of the translation work. That is why the translating companies aim at the highest possible matches. Such result can be achieved with bigger translation memories which have higher coverage and it is also the aim of this paper.

The results deserve more detailed description. The analysis table columns are: **Match** – type of match between TM^s and D^s, **Seg** – number of segments identified in D^s, **wrds** – number of source words which are covered (translatable) by the TM^s, **chars** – number of source characters and **%** – percentage of coverage for the type of match in the first column.

In the evaluation process, we have tested the translation on a document with 4,563 segments, 35,142 words and 211,407 characters.

In the measurements, we have split the analysis for the subsegment combination method to the values obtained by a) the TM_{sub} translation memory of subsegments (consistent MT phrases from OPUS), i.e. without the *join* operation, see Table 1, and b) the $TM_{subjoin}$ memory further expanded by means of

the *join* operation,[4] see Table 2. The results in both tables display three subtables: the original analysis with the TM^s input translation memory, the analysis using only the new $\mathsf{TM}_{sub}/\mathsf{TM}_{subjoin}$ translation memory, and the most practical combination of $\mathsf{TM}^s + \mathsf{TM}_{sub}/\mathsf{TM}_{subjoin}$.

The most important are the boldface numbers at the bottom of the tables expressing the sum percentage of any of the translation match. The result in Table 1 is 15.64%. This means that with the TM_{sub} we can increase the coverage of TM^s by more than 9% which is a substantial improvement over using TM^s alone.

The results of the *join* operation in Table 2 further increase the total percentage of matches to 16.15%. When compared to the TM_{sub} results, this represents quite low improvement of 0.5%. The problem currently lies in the coverage of the current prototype implementation of the *join* operation. In the evaluated document, the $\mathsf{TM}_{subjoin} - \mathsf{TM}_{sub}$ phrases cover only 122 subsegments of the document, which is too low to generate a substantial increase in translation matches. With regard to the coverage, the subsegment lexicalization method B should also provide more interesting results. This remains, however, still to be implemented and evaluated.

5 Conclusion

In this paper, we have described several novel methods for expanding translation memories. We showed that we can effectively generate new high quality translation pairs which increase the efficiency of computer-aided translation by means of purely computational linguistically motivated techniques.

The presented results show and improvement of 10 percent in the translation matches, which already corresponds to substantial economic savings in the translation process.

In the future work, we will concentrate on the evaluation of the other presented methods and their application in a selected CAT system.

Acknowledgements This work has been partly supported by the Ministry of Education of CR within the LINDAT-Clarin project LM2010013.

References

1. Tiedemann, J.: Parallel Data, Tools and Interfaces in OPUS. In: Proceedings of the Eighth International Conference on Language Resources and Evaluation, LREC 2012. (2012) 2214–2218 `http://opus.lingfil.uu.se`.
2. Koehn, P.: Europarl: A parallel corpus for statistical machine translation. In: MT summit. Volume 5. (2005) `http://www.statmt.org/europarl`.
3. Steinberger, R., Pouliquen, B., Widiger, A., Ignat, C., Erjavec, T., Tufis, D., Varga, D.: The JRC-Acquis: A multilingual aligned parallel corpus with 20+ languages. arXiv preprint cs/0609058 (2006)

[4] i.e. $\mathsf{TM}_{sub} \subset \mathsf{TM}_{subjoin}$

4. Trombetti, M.: Creating the world's largest translation memory. In: MT Summit. (2009) http://mymemory.translated.net.
5. Planas, E., Furuse, O.: Formalizing translation memories. In: Machine Translation Summit VII. (1999) 331–339
6. Planas, E., Furuse, O.: Multi-level similar segment matching algorithm for translation memories and example-based machine translation. In: Proceedings of the 18th conference on Computational linguistics-Volume 2, Association for Computational Linguistics (2000) 621–627
7. Désilets, A., Farley, B., Stojanovic, M., Patenaude, G.: WeBiText: Building large heterogeneous translation memories from parallel web content. Proc. of Translating and the Computer **30** (2008) 27–28
8. Nevado, F., Casacuberta, F., Landa, J.: Translation memories enrichment by statistical bilingual segmentation. In: Proceedings of the Fourth International Conference on Language Resources and Evaluation, LREC 2004. (2004)
9. Koehn, P., Och, F.J., Marcu, D.: Statistical phrase-based translation. In: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1, Association for Computational Linguistics (2003) 48–54
10. Simard, M., Langlais, P.: Sub-sentential exploitation of translation memories. In: Machine Translation Summit VIII. (2001) 335–339
11. Macklovitch, E., Simard, M., Langlais, P.: TransSearch: A Free Translation Memory on the World Wide Web. In: Proceedings of the Second International Conference on Language Resources and Evaluation, LREC 2000. (2000)
12. Och, F.J., Ney, H.: A systematic comparison of various statistical alignment models. Computational linguistics **29**(1) (2003) 19–51
13. Och, F.J.: Giza++ software. http://www.statmt.org/moses/giza/GIZA++.html (2003)