Increasing Coverage of Translation Memories with Linguistically Motivated Segment Combination Methods

Vít Baisa, Aleš Horák, Marek Medved'
Natural Language Processing Centre, Masaryk University, Brno, Czech Republic
{xbaisa,hales,xmedved1}@fi.muni.cz

Abstract

Translation memories (TMs) used in computer-aided translation (CAT) systems are the highest-quality source of parallel texts since they consist of segment translation pairs approved by professional human translators. The obvious problem is their size and coverage of new document segments when compared with other parallel data.

In this paper, we describe several methods for expanding translation memories using linguistically motivated segment combining approaches concentrated on preserving the high translational quality. The evaluation of the methods was done on a medium-size real-world translation memory and documents provided by a Czech translation company as well as on a large publicly available DGT translation memory published by European Commission. The asset of the TM expansion methods were evaluated by the pre-translation analysis of widely used MemoQ CAT system and the METEOR metric was used for measuring the quality of fully expanded new translation segments.

1 Introduction

Most professional translators use a specific CAT system with provided or self-built translation memories (TM). The translation memories are usually in-house, costly and manually created resources of varying sizes of thousands to millions of translation pairs.

Only recently some TMs have been made publicly available: DGT (Steinberger et al., 2013), or MyMemory (Trombetti, 2009); to mention just a few. But there is still a heavy demand on enlarging and improving TMs and their coverage of new input documents to be translated.

Obviously, the aim is to have a TM that best fits to the content of a new document as this is crucial for speeding up the translation process: when a larger part of a document can be pre-translated by a CAT system, the translation itself can be cheaper. Coverage of TMs is directly translatable to savings by translation companies and their customers.

We present two methods for expanding TMs: subsegment generation and subsegment combination. The idea behind these methods is based on the fact, that even if the topic of the new document is well covered by the memory, only very rarely the memory includes exact sentences (segments) as they appear in the document. The differences between known and new segments often consist of substitutions or combinations of particular known subsegments.

The presented methods concentrate on increasing the coverage of the content of an existing TM with regard to a new document, and at the same time try to keep a reasonable quality of newly generated segment pairs. We work with English-Czech data but the procedures are mostly language independent.

Evaluation was done on several documents and a medium-size in-house translation memory provided by a large Czech translation company. For comparison, we have also tested the methods on the DGT translation memory.

2 Related work

Translation memories are generally understudied within the field of NLP. Machine translation techniques, especially *example-based machine translation* (*EBMT*) employ translation memories in an approach similar to CAT systems (Planas and Furuse, 1999) but NLP approaches have not been applied on them extensively.

TM-related papers mainly focus on algorithms for searching, matching and suggesting segments

CZ	EN	probabilities				alignment	
být větší	be greater	0.538	0.053	0.538	0.136	0-0	1-1
být větší	be larger	0.170	0.054	0.019	0.148	0-0	1-1

Figure 1: An example of generated subsegments – consistent phrases

within CAT systems (Planas and Furuse, 2000).

In (Désilets et al., 2008), 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 they developed system *WeBiText* for extracting possible segments and their translations from bilingual web pages.

In the study (Nevado et al., 2004), the authors exploited two methods of segmentation of translation memories. Their approach starts with a similar assumption as our subsegment combination methods presented below, i.e. that a TM coverage can be increased by splitting the TM segments to smaller parts (subsegments). In both cases, the subsegments are generated via the phrase-based machine translation (PBMT) technique (Koehn et al., 2003). However, our methods do not present the subsegments as the results. The subsegments are used in segment combination methods to obtain new larger translational phrases, or full segments in the best case.

(Simard and Langlais, 2001) describes a method of sub-segmenting translation memories which deals with the principles of EBMT. The authors of this study created an on-line system TransSearch (Macklovitch et al., 2000) 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.

3 Subsegment generation

In the first step, the proposed TM expansion methods process the available translation memory and generate all *consistent phrases* as subsegments from it. Subsegments and the corresponding translations are generated using the Moses (Koehn et al., 2007) tool directly from the TM, no additional data is used.

The word alignment is based on MGIZA++ (Gao and Vogel, 2008) (parallel version of GIZA++ (Och and Ney, 2003)) and the default Moses heuristic *grow-diag-final*.[1] The next steps are phrase extraction and scoring (Koehn et al., 2007). The corresponding extended TM is denoted as SUB. The output from subsegment generation contains for each subsegment its translation, probabilities and alignment points, see Figure 1 for an example.

The four probabilities are *inverse phrase translation probability*, *inverse lexical weighting*, *direct phrase translation probability* and *direct lexical weighting*, respectively. They are obtained directly from Moses procedures. These probabilities are used to select the best translations in case there are multiple translations for a subsegment. Alternative translations for a subsegment are combined from different aligned pairs in the TM. Typically, short subsegments have many translations.

The alignment points determine the word alignment between subsegment and its translation, i.e. *0-0 1-1* means that the first word *být* from source language is translated to the first word in the translation *be* and the second word *větší* to the second *greater*. These points give us important information about the translation: 1) empty alignment, 2) one-to-many alignment and 3) opposite orientation.

4 Subsegment combination

The output of the subsegment generation is denoted as a special translation memory named SUB. The obtained subsegments are then filtered and used by the following methods for subsegment combination with regard to the segments from the input document:

- JOIN: new segments are built by concatenating two segments from SUB, output is J.
    1. JOIN:O: joint subsegments overlap in a segment from the document, output=OJ.
    2. JOIN:N: joint subsegments neighbour in a segment from the document, output=NJ.

[1] `http://www.statmt.org/moses/?n=FactoredTraining.AlignWords`

Table 1: An example of the SUBSTITUTE:O method, Czech → English.

original subsegments	• “lze rozdělit do těchto kategorií:” (*can be divided into these categories:*) • “následujících kategorií” (*the following categories*)
new subsegment	“lze rozdělit do následujících kategorií:”
its translation	*can be divided into the following categories:*

- SUBSTITUTE: new segments can be created by replacing a part of one segment with another subsegment from SUB, output is S.

  1. SUBSTITUTE:O: the gap in the first segment is covered with an overlap with the second subsegment, see the example in Table 1, output is OS.
  2. SUBSTITUTE:N: the second subsegment is inserted into the gap in the first segment, output is NS.

During the subsegment non-overlapping combination, the acceptability of the combination is decided (and ordered) by measuring a language fluency score obtained by a combined n-gram score (for $n = \langle 1..5 \rangle$) from a target language model.[2] The quality of the subsegment translation can be increased by filtering the used subsegments on noun phrase boundaries.

The algorithm for the JOIN method actually works with indexes which represent the subsegment positions in the tokenized segment. The available subsegments are processed as a list I ordered by the subsegment size (in the number of tokens, in descending order). The process starts with the biggest subsegment in the segment and then tries to join it successively with other subsegments. If it succeeds, the new subsegment is appended to a temporary list T. After all other subsegments are processed, the temporary list T of new subsegments is prepended to I and the algorithm starts with a new subsegment created from the two longest subsegments. If it does not succeed, the next subsegment in the order is processed. The algorithm thus prefers to join longer subsegments. In each iteration it generates new (longer) subsegments and it discards one processed subsegment.

5 Evaluation

For the evaluation of the proposed methods, we have used a medium-size in-house translation memory provided by a Czech translation company and two real-world documents of nearly 5,000 segments with their referential translations. The TM contains 144,311 Czech-English translation pairs filtered from the complete company's TM by the same topic as the tested documents. For a comparison, we have run and evaluated the methods also on publicly available DGT translation memory (Steinberger et al., 2013) with the size over 300,000 translation pairs.

For measuring coverage of the expanded TMs we have used the document and TM analysis tool included in the MemoQ software. The same evaluation is used by translation companies for an assessment of the actual translation costs. The results have been obtained directly from the pre-translation analysis of the MemoQ system. The results are presented in Table 2. The TM column contains the results for the original non-expanded translation memory. The column SUB displays the analysis for subsegments (consistent phrases) derived from the original TM. The other columns correspond to the methods JOIN, see Section 4. The final column “all” is the resulting expanded TM obtained as a combination of all tested methods. All numbers represent coverage of segments from the input document versus segments from expanded TMs. The analysis divides all matches segments to categories (lines in the tables. Each category denotes how many words from the segment were found in the analysed TM. 100% match corresponds to the situation when a whole segment from D can be translated using a segment from the respective TM. Translations of shorter parts of the segment are then matches lower than 100%. The most valuable matches for translation companies and translators are those over 75–85%. The presented results show an analysis of the expanded TM for documents with 4,563 segments (35,142 words and 211,407 characters).

[2] In current experiments, we have trained a language model using KenLM (Heafield, 2011) tool on first 50 million sentences from the enTenTen corpus (Jakubíček et al., 2013).

Table 2: MemoQ analysis, TM, coverage in %.

Match	TM	SUB	OJ	NJ	all
100%	0.41	0.12	0.10	0.17	**0.46**
95–99%	0.84	0.91	0.64	0.90	1.37
85–94%	0.07	0.05	0.25	0.76	0.81
75–84%	0.80	0.91	1.71	3.78	4.40
50–74%	8.16	10.05	25.09	40.95	42.58
any	10.28	12.04	27.79	46.56	**49.62**

Table 3: MemoQ analysis, DGT-TM.

Match	SUB	OJ	NJ	all
100%	0.08	0.07	0.11	**0.28**
95–99%	0.75	0.44	0.49	0.66
85–94%	0.05	0.08	0.49	0.61
75–84%	0.46	0.96	3.67	3.85
50–74%	10.24	27.77	41.90	44.47
all	11.58	29.32	46.66	**49.87**

For a comparison we also tested the methods on DGT translation memory (Steinberger et al., 2013). We have used 330,626 pairs from 2014 release. See Table 3 for the results of DGT alone and Table 4 for combination of the TM and DGT.

Table 4: MemoQ analysis, TM + DGT-TM.

Match	SUB	OJ	NJ	all
100%	0.15	0.13	0.29	**0.57**
95–99%	0.98	0.59	1.24	1.45
85–94%	0.09	0.22	1.34	1.37
75–84%	1.03	2.26	6.35	7.07
50–74%	12.15	34.84	49.82	51.62
all	14.40	38.04	59.04	**61.51**

Table 5: METEOR, 100% matches company in-house translation memory

feature	SUB	OJ	NJ	NS
prec	0.60	0.63	0.70	0.66
recall	0.67	0.74	0.74	0.71
F1	0.64	0.68	0.72	0.68
METEOR	0.31	0.37	**0.38**	**0.38**
DGT				
prec	0.76	0.93	0.91	0.81
recall	0.78	0.86	0.88	0.85
F1	0.77	0.89	0.89	0.83
METEOR	0.40	0.50	**0.51**	0.45

Table 6: Error examples, Czech → English.

source seg.	Oblast dat může mít libovolný tvar.
reference	The data area may have an arbitrary shape.
generated seg.	Area data may have any shape.

We have also compared the results with the output of a function called *Fragment assembly* (Teixeira, 2014), that is present in the MemoQ CAT system.[3] Fragment assembly suggests new segments based on several dictionary and non-word elements (term base, non-translatable hits, numbers, auto-translatable hits). Unknown subsegments are taken from the source language in the tested setup. For measuring the quality of translation (accuracy), we have used METEOR metric (Denkowski and Lavie, 2014). We have achieved score 0.29 with our data in comparison with MemoQ CAT system with score 0.03 when computed for all segments including those with empty translations to the target language. When we take into account just the segments that are pre-translated by MemoQ Fragment assembly as well as by our methods (871 segments), we have achieved the score of 0.36 compared to 0.27 of MemoQ. As the METEOR evaluation metric has been proposed to evaluate MT systems, it assumes that we have fully translated segments (pairs). We have thus provided a "mixed" translation in the same way as it is done in the MemoQ Fragment assembly technique – non-translated phrases (subsegments) appear in the output segment "as is", i.e. in the source language. The resulting segment can thus be a combination of source and target language words, which is correspondingly taken into account by the METEOR metric. We have also measured the asset of particular methods with regard to the translation quality, however, in this case we have measured just full 100% matched segments. The results are presented in Table 5. Nevertheless this evaluation was done for the sake of completeness. It is well known that automatic evaluation metrics for assessing machine translation quality are not fully reliable and that a human evaluation is always needed.

Error analysis Regarding the precision we have analysed some problematic cases. The most common error was when subsegments are combined in the order in which they occur in the segment assuming the same order in a target language, see the Table 6.

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

We plan to include a phrase assembly technique that would analyse the input noun phrases and test the fluency of their translation by means of the language model. Results that would not pass a threshold will not take part in the final segment combination method. The best evaluation would be extrinsic: to use generated TMs in a process of translation of a set of documents and measure time needed for the translation.

6 Conclusion

We presented two methods JOIN and SUBSTITUTE which generate new segment pairs for any translation memory and input document. Both methods have variants with overlap and adjoint segments. The techniques include linguistically motivated techniques for filtering out phrases, which provide non-fluent output texts in the target language.

We are co-operating with one of major Central-European translation company which provided us with the testing data and we plan to deploy the methods in their translation process within a future project.

Acknowledgments

This work has been partly supported by the Ministry of Education of CR within the LINDAT-Clarin project LM2010013 and by the Grant Agency of CR within the project 15-13277S.

References

Michael Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the EACL 2014 Workshop on Statistical Machine Translation*.

Alain Désilets, Benoit Farley, M Stojanovic, and G Patenaude. 2008. WeBiText: Building large heterogeneous translation memories from parallel web content. *Proc. of Translating and the Computer*, 30:27–28.

Qin Gao and Stephan Vogel. 2008. Parallel implementations of word alignment tool. In *Software Engineering, Testing, and Quality Assurance for Natural Language Processing*, pages 49–57. Association for Computational Linguistics.

Kenneth Heafield. 2011. Kenlm: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197. Association for Computational Linguistics.

Miloš Jakubíček, Adam Kilgarriff, Vojtěch Kovář, Pavel Rychlý, Vít Suchomel, et al. 2013. The tenten corpus family. In *Proc. Int. Conf. on Corpus Linguistics*.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. 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*, pages 48–54. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, pages 177–180. Association for Computational Linguistics.

Elliott Macklovitch, Michel Simard, and Philippe Langlais. 2000. TransSearch: A Free Translation Memory on the World Wide Web. In *Proceedings of the Second International Conference on Language Resources and Evaluation, LREC 2000.*

Francisco Nevado, Francisco Casacuberta, and Josu Landa. 2004. Translation memories enrichment by statistical bilingual segmentation. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation, LREC 2004.*

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational linguistics*, 29(1):19–51.

Emmanuel Planas and Osamu Furuse. 1999. Formalizing translation memories. In *Machine Translation Summit VII*, pages 331–339.

Emmanuel Planas and Osamu Furuse. 2000. 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*, pages 621–627. Association for Computational Linguistics.

Michel Simard and Philippe Langlais. 2001. Sub-sentential exploitation of translation memories. In *Machine Translation Summit VIII*, pages 335–339.

Ralf Steinberger, Andreas Eisele, Szymon Klocek, Spyridon Pilos, and Patrick Schlüter. 2013. Dgt-tm: A freely available translation memory in 22 languages. *arXiv preprint arXiv:1309.5226.*

Carlos SC Teixeira, 2014. *The Handling of Translation Metadata in Translation Tools*, page 109. Cambridge Scholars Publishing.

Marco Trombetti. 2009. Creating the world's largest translation memory. In *MT Summit*. `http://mymemory.translated.net`.