# Improving Coverage of Translation Memories with Language Modelling

Vít Baisa, Josef Bušta, and Aleš Horák

Natural Language Processing Centre
Faculty of Informatics, Masaryk University
Botanická 68a, 602 00 Brno, Czech Republic
{xbaisa,xbusta1,hales}@fi.muni.cz

**Abstract.** In this paper, we describe and evaluate current improvements to methods for enlarging translation memories. In comparison with the previous results in 2013, we have achieved improvement in coverage by almost 35 percentage points on the same test data. The basic subsegment splitting of the translation pairs is done using Moses and (M)GIZA++ tools, which provide the subsegment translation probabilities. The obtained phrases are then combined with subsegment combination techniques and filtered by large target language models.

**Keywords:** translation memory, CAT, segment, subsegment leveraging, partial translation, Moses, GIZA++, word matrix, METEOR, MemoQ, language model

## 1 Introduction

Computer-aided translation (CAT) is becoming more and more popular—with the state-of-the-art technologies such as subsegment leveraging, machine translation, or automatic terminology extraction, the translation process is faster and easier than ever before.

CAT systems depend on translation memories: manually built databases of aligned source and target segments (phrases, sentences, paragraphs). They can be considered as parallel corpora of very high-quality (since they are prepared by professional translators) but of quite small size and coverage of new documents.

We describe current improvements of the methods for expanding translation memories which have been described in the previous paper [1]. The goal of these methods is to increase new document coverage of a translation memory preserving its high translational precision.

There is also a commercial aspect of this research: the coverage analyses provided by CAT systems are usually used for estimating the amount of work needed for translating a given document (i.e. the price of the translation work). The higher number of segments which can be pre-translated automatically, the lower is the price of the translation work. That is why the translation (and localization) companies aim at the highest coverage of their resources.

Aleš Horák, Pavel Rychlý (Eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2014, pp. 27–34, 2014. 

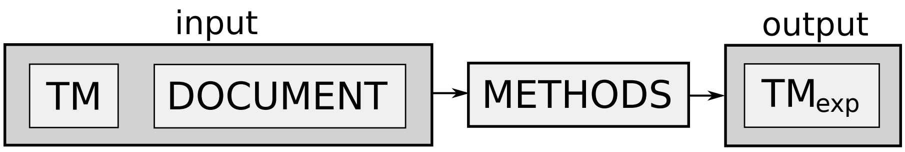


Fig. 1: Schema of the basic work flow for $\mathsf{TM}_{exp}$.

## 2 Previous and Related Work

In the previous paper [1], we have proposed several methods for enlarging translation memories and provided an evaluation for one of them. In this paper, we describe the improvements of the methods and evaluate all of them on both the original data used in the previous paper and also on a new data, Directorate-General for Translation or DGT[1] [2] translation memory released recently by the European Commission. For related work refer to [1].

## 3 Subsegment Processing Methods

In this section, we present the changes and improvements to the previous paper [1] and a detailed description of the implemented techniques.

The input for our methods is a translation memory and a document. We want to enlarge the TM (the expanded TM is denoted $\mathsf{TM}_{exp}$) to cover more segments in the document and preserve the quality of the translations, see the Figure 1.

### 3.1 Method A: Subsegment Generation

Subsegments and the corresponding translations are generated using Moses [3] tool directly from the TM, no additional data is used. The word alignment is based on MGIZA++ [4] (parallel version of GIZA++ [5]) and the default Moses heuristic *grow-diag-final*.[2] The next steps are phrase extraction and scoring [3]. The corresponding partially expanded TM is denoted as $\mathsf{TM}^{sub}$. The output from subsegment generation has the following format:

| **Subsegment** | **Translation** | **Probabilities** | **Alignment points** |
|---|---|---|---|
| nejlepší uhlí | best coal | 0.158, 0.142, 0.158, 0.69 | 0-0 1-1 |

The probabilities are *inverse phrase translation probability*, *inverse lexical weighting*, *direct phrase translation probability* and *direct lexical weighting* obtained directly from the Moses procedures. These probabilities are used to select the best

[1] `https://ec.europa.eu/jrc/en/language-technologies/dgt-translation-memory`

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

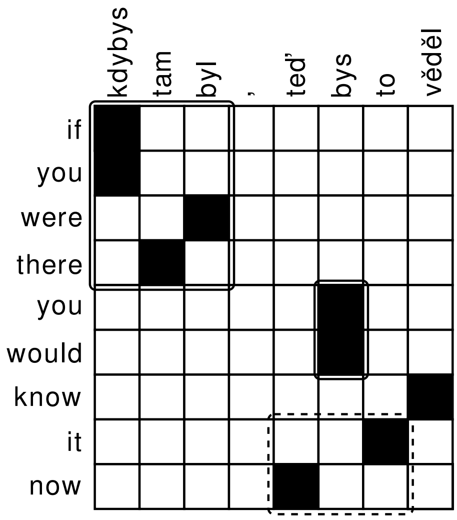


Fig. 2: Word matrix for two aligned sentences / segments.

translations in case there are many 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 "*nejlepší*" from the source language is translated to the first word in the translation "*best*" and the second word "*uhlí*" to the second word "*coal*." These points give us an important information about the subsegment translation: 1) empty alignment, 2) one-to-many alignment, and 3) opposite orientation.

In Figure 2 the empty alignment is represented by an empty line or an empty row, the one-to-many alignment by a sequence of adjacent squares in a row or in a column and the opposite orientation by a sequence of neighbouring squares on the secondary diagonal. The alignments are used to determine correct positions in the subsegments translations.

### 3.2 Method B: Subsegment Combination

The subsegment translation pairs obtained by the method A are used as a pool of candidate subsegments used in the next method to generate longer subsegments. In an ideal case, to generate a new translation pair covering a whole, originally uncovered, segment in the input document – so called *100 % match*.

Currently, the sub-methods *join* and *substitute* are proposed for subsegment combinations, each of them in an *overlapping* and *non-overlapping* variant:

1. JOIN: new segments are built by concatenating two segments from $\mathrm{TM}^{sub}$, denoted $\mathrm{TM}^{J}$.
   (a) $\mathrm{JOIN}^{O}$: joined subsegments overlap in a segment from the document, denoted $\mathrm{TM}^{OJ}$.

Table 1: SUBSTITUTE$^O$, example for Czech → English

| | |
|---|---|
| new subsegment | Provozovatelé musí dodržovat zvláštní pravidla pro výzkumné |
| its translation | Operators shall comply with the special rules on research |
| from subsegments | Provozovatelé **musí** vytvářet **zvláštní** pravidla pro výzkumné \| **musí** dodržovat **zvláštní** |
| their translations | Operators **shall** create **the special** rules on research \| **shall** comply with **the special** |

   (b) JOIN$^N$: joined subsegments neighbour in a segment from the document, denoted TM$^{NJ}$.

2. SUBSTITUTE: new segments can be created by replacing a part of one segment with another subsegment from TM$^{sub}$, denoted TM$^S$.
   (a) SUBSTITUTE$^O$: the gap in the first segment is covered with an overlap with the second subsegment, see the example in Table 1, denoted TM$^{OS}$.
   (b) SUBSTITUTE$^N$: the second subsegment is inserted into the gap in the first segment, denoted TM$^{NS}$.

During the subsegment non-overlapping combination, any two subsegments are combined regardless the fluency and the context. That is why we need to evaluate the quality of the combination. For the quality measurement, we have trained a language model using KenLM [6] tool on first 50 million sentences from enTenTen [7] with model order set to 5.

The translation quality of the SUBSTITUTE operation can be improved by substituting a particular part-of-speech (noun, adjective, ...) for the same part-of-speech or a noun phrase for a noun phrase.

**Algorithm 1:** JOIN subsegments

**Data**: Segment $S$ from document; List $I$ of indexes $(i,\ j)$ of subsegments occurring in $S$ sorted in decreasing order by the difference of $j - i$
**Result**: $R$

**while** $I \neq \emptyset$ **do**
  $(i, j) \longleftarrow$ `First`$(I)$;
  $I \longleftarrow I - (i, j)$;
  $T \longleftarrow \emptyset$;
  **for** $(k,\ l) \in I$ **do**
    **if** $(k < i \wedge l + 1 \geq i \wedge j > l) \vee (i < k \wedge j + 1 \geq k \wedge l > j)$ **then**
      $T \longleftarrow T +$ (`Min` $(k, i)$,`Max` $(l, j)$);
      $R \longleftarrow R +$ (`Min` $(k, i)$,`Max` $(l, j)$);
      **if** (`Min` $(k, i)$,`Max` $(l, j)$) $=$ (0,`Length` $(S)$) **then**
        **return** $R$;
  $I \longleftarrow T + I$;
**return** $R$;

In [1], the operation JOIN was implemented just for non-overlapping subsegments and as a concatenation of any two subsegments. In this paper, we present an improved Algorithm 1. The algorithm works with indexes which represent the subsegment positions in the tokenized segment from the input document. The processing starts with the biggest subsegment in the segment and then tries to join it with other subsegments. If it succeeds, the new subsegment is appended to temporary list T. After all other subsegments are processed, T 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 1 prefers to join longer subsegments. In each iteration it generates new (longer) subsegments and it discards one processed subsegment. See Section 4 for the evaluation of this new approach.

## 4 Evaluation

For the evaluation of the current implementation of the TM-expanding methods, we have used the same translation memory $TM^s$ and the same example document $D^s$ as in [1]. Both data files have been provided by one of the biggest Czech translation companies.

Table 2: MemoQ analysis for $TM^s$.

| | TM | | | | $TM^{sub}$ | | | | $TM^{NS}$ | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Match** | **Seg** | **wrds** | **chars** | **%** | **Seg** | **wrds** | **chars** | **%** | **Seg** | **wrds** | **chars** | **%** |
| 100% | 23 | 128 | 813 | 0.4 | 165 | 178 | 611 | 0.51 | 0 | 0 | 0 | 0 |
| 95–99% | 45 | 185 | 1,130 | 0.5 | 193 | 245 | 1,578 | 0.7 | 20 | 43 | 273 | 0.12 |
| 85–94% | 4 | 21 | 155 | 0.1 | 19 | 50 | 325 | 0.14 | 18 | 78 | 451 | 0.22 |
| 75–84% | 42 | 208 | 1,305 | 0.6 | 96 | 310 | 1,888 | 0.88 | 129 | 436 | 2,677 | 1.24 |
| 50–74% | 462 | 1,689 | 10,293 | 4.8 | 789 | 4,543 | 27,999 | 12.93 | 1681 | 12,522 | 75,108 | 35.65 |
| ≥ 75% | 114 | 542 | 3403 | 1.6 | 473 | 783 | 4,402 | 2.23 | 167 | 557 | 3,401 | 1.58 |
| any | 576 | 2,231 | 13,696 | 6.4 | 1,262 | 5,326 | 32,401 | 15.16 | 1,848 | 13,079 | 78,509 | 37.23 |

| | $TM^{OJ}$ | | | | $TM^{NJ}$ | | | | $TM^{all}$ | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Match** | **Seg** | **wrds** | **chars** | **%** | **Seg** | **wrds** | **chars** | **%** | **Seg** | **wrds** | **chars** | **%** |
| 100% | 6 | 23 | 106 | 0.07 | 4 | 19 | 101 | 0.05 | 182 | 302 | 1360 | **0.86** |
| 95–99% | 11 | 60 | 310 | 0.17 | 13 | 87 | 466 | 0.25 | 232 | 465 | 2,858 | 1.32 |
| 85–94% | 5 | 33 | 217 | 0.09 | 17 | 149 | 892 | 0.42 | 41 | 221 | 1,382 | 0.63 |
| 75–84% | 68 | 314 | 1,809 | 0.89 | 110 | 881 | 5,022 | 2.51 | 265 | 1,475 | 8,655 | 4.2 |
| 50–74% | 1,153 | 7,667 | 45,641 | 21.83 | 1,354 | 11,997 | 70,730 | 34.15 | 1,507 | 15,324 | 92,158 | 43.62 |
| ≥ 75% | 90 | 430 | 2,442 | 1.22 | 144 | 1,136 | 6,481 | 3.23 | 720 | 2,463 | 14,255 | 7.01 |
| any | 1,243 | 8,097 | 48,083 | 23.05 | 1,498 | 13,133 | 77,211 | 37.38 | 2,227 | 17,787 | 106,413 | **50.63** |

Table 3: MemoQ analysis for DGT-TM.

| | TM | | | | TM^sub | | | | TM^NS | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Match** | **Seg** | **wrds** | **chars** | % | **Seg** | **wrds** | **chars** | % | **Seg** | **wrds** | **chars** | % |
| 100% | 31 | 59 | 639 | 0.03 | 276 | 457 | 2,666 | 0.25 | 58 | 260 | 953 | 0.45 |
| 95–99% | 198 | 546 | 1,941 | 0.30 | 225 | 446 | 1,998 | 0.24 | 206 | 827 | 2,992 | 0.45 |
| 85–94% | 43 | 169 | 986 | 0.09 | 208 | 971 | 4,205 | 0.53 | 94 | 492 | 2,187 | 0.27 |
| 75–84% | 357 | 1,745 | 8,021 | 0.96 | 386 | 1,714 | 9,115 | 0.94 | 287 | 1,492 | 7,102 | 0.82 |
| 50–74% | 2,580 | 20,778 | 126,273 | 11.37 | 2,907 | 22,736 | 141,526 | 12.45 | 3,348 | 29,549 | 182,667 | 16.18 |
| ≥ 75% | 629 | 2,519 | 11,587 | 1.38 | 1,095 | 3,588 | 17,984 | 1.96 | 645 | 3,071 | 13,234 | 1.99 |
| any | 3,209 | 23,297 | 137,860 | 12.75 | 4,002 | 26,324 | 159,510 | 14.41 | 3,993 | 32,620 | 195,901 | 17.86 |

| | TM^OJ | | | | TM^NJ | | | | TM^all | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Match** | **Seg** | **wrds** | **chars** | % | **Seg** | **wrds** | **chars** | % | **Seg** | **wrds** | **chars** | % |
| 100% | 38 | 187 | 764 | 0.10 | 29 | 161 | 683 | 0.09 | 358 | 838 | 4,172 | **0.46** |
| 95–99% | 195 | 770 | 2,752 | 0.42 | 69 | 247 | 769 | 0.14 | 338 | 990 | 4,282 | 0.54 |
| 85–94% | 124 | 695 | 3,198 | 0.38 | 203 | 1,107 | 4,892 | 0.61 | 133 | 666 | 3,750 | 0.36 |
| 75–84% | 256 | 1,634 | 7,764 | 0.89 | 287 | 2,133 | 10,331 | 1.17 | 537 | 3,231 | 17,340 | 1.77 |
| 50–74% | 3,220 | 32,325 | 200,667 | 17.70 | 3,673 | 47,715 | 298,031 | 26.12 | 4,183 | 53,791 | 343,699 | 29.45 |
| ≥ 75% | 613 | 3,286 | 14,478 | 1.79 | 588 | 3,648 | 16,675 | 2.01 | 1,366 | 5,725 | 29,544 | 3.13 |
| any | 3,833 | 35,611 | 215,145 | 19.49 | 4,261 | 51,363 | 314,706 | 28.13 | 5,549 | 59,516 | 373,243 | **32.58** |

The evaluation results have been obtained directly from the pre-translation analysis of the MemoQ[3] system. The statistics express how many segments from the document $D^s$ can be translated automatically using the TM-expanding methods. The automatic translation is done on the segment level and even on lower levels of subsegments. The partial matches are expressed as the match percentages in the table. The 100% match corresponds to the situation when a whole segment from $D^s$ can be translated using a segment from the respective translation memory (either the original one or a memory obtained by each particular sub-method). Translations of shorter parts of the segment are then matches lower than 100%.

The columns in Tables 2 and 3 are: **Match**: type of match between TM and $D^s$, **Seg**: number of segments identified in $D^s$, **wrds**: number of source words which are covered (translatable) by TM, **chars**: number of source characters, and percent sign: percentage of coverage for the type of match in the first column. In the evaluation process, we have first tested the translation on a document with 4,563 segments (35,142 words and 211,407 characters), see Table 2.

For an independent comparison, we also present our results for DGT translation memory [2]. For the evaluation using DGT we have used 330,626 pairs from 2014 release and evaluated it on 10,000 randomly chosen segments from the same release. Duplicate pairs were removed before evaluation. See Table 3 for the results.

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

Table 4: Analysis of dependence between subsegment length and the coverage of the document.

| | TM^s | | | | | DGT-MT | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Length** | $TM^{sub}$ | $TM^{OJ}$ | $TM^{NJ}$ | $TM^{NS}$ | $TM^{all}$ | $TM^{sub}$ | $TM^{OJ}$ | $TM^{NJ}$ | $TM^{NS}$ | $TM^{all}$ |
| $\geq 1$ | 85% | 11% | 15% | 25% | 85% | 95% | 57% | 71% | 65% | 95% |
| $\geq 2$ | 35% | 11% | 15% | 25% | 44% | 78% | 57% | 71% | 65% | 85% |
| $\geq 3$ | 7% | 11% | 15% | 25% | 32% | 53% | 57% | 71% | 65% | 82% |
| $\geq 4$ | 1% | 5% | 15% | 9% | 16% | 35% | 52% | 71% | 53% | 74% |
| $\geq 5$ | 0% | 2% | 8% | 2% | 7% | 23% | 45% | 66% | 38% | 65% |

Table 5: Translation quality (METEOR score) for 100% matches.

| | TM^s | | | | | DGT-MT | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **feature** | $TM^{sub}$ | $TM^{OJ}$ | $TM^{NJ}$ | $TM^{NS}$ | $TM^{all}$ | $TM^{sub}$ | $TM^{OJ}$ | $TM^{NJ}$ | $TM^{NS}$ | $TM^{all}$ |
| precision | 0.60 | 0.63 | 0.70 | 0.66 | 0.61 | 0.76 | 0.93 | 0.91 | 0.81 | 0.80 |
| recall | 0.67 | 0.74 | 0.74 | 0.71 | 0.68 | 0.78 | 0.86 | 0.88 | 0.85 | 0.81 |
| f1 | 0.64 | 0.68 | 0.72 | 0.68 | 0.64 | 0.77 | 0.89 | 0.89 | 0.83 | 0.81 |
| METEOR score | 0.31 | 0.37 | **0.38** | **0.38** | 0.31 | 0.40 | 0.50 | **0.51** | 0.45 | 0.43 |

We have also counted the coverage of the document considering the length of subsegments, see Table 4. Notice that longer subsegments are created by subsegment combination.

The METEOR [8] metric was used to evaluate quality (precision) of the proposed translated segments. We provide statistics for all implemented methods on both test data sets, see Table 5. The METEOR evaluation metric has been proposed to evaluate MT systems, therefore it assumes that we have fully translated segments (pairs). That is why we are evaluating only 100% matches since it is not straightforward to interpret METEOR scores for partially translated candidate sentences.

We have analysed the problematic cases regarding the precision. The most common error is when subsegments are combined in the order in which they occur in the segment assuming the same text sequential order in the target language, see the Table 6. We assume, that such errors will be less frequent with a larger input translation memory, which will offers higher ration of the overlapped (contextual) segments.

Table 6: Non-overlapping JOIN error example Czech → English.

| | |
|---|---|
| segment | Prémie na bramborový škrob |
| reference | Potato starch premium |
| new subsegment | Prémie na bramborový škrob |
| its translation | Premiums potato starch |
| from subsegments | Prémie na \| bramborový škrob |
| their translations | Premiums \| potato starch |

## 5 Conclusions

We have shown that the originally proposed methods can be further improved and provided the evaluation which shows that the coverage of all matches has been increased by 34.5 percentage points (from 16.15% reported in [1] to 50.63%). As for the 100% matches which are the most important, the test results show an increase of 0.5 percentage points comparing the original TM and combination of both JOIN methods (from 0.4% reported earlier to 0.86%) and the coverage of $> 75\%$ matches increased by 5.4% (from 1.6% to 7%).

The translational quality of the resulting new segments is kept at the high level as is shown by the METEOR score up to 0.51 for the evaluation with the translation memory by Directorate-General for Translation (DGT) of the European Commission.

**Acknowledgements** The work has been partly supported by the OP VaVpI project No CZ.1.05/3.1.00/10.0216.

## References

1. Baisa, V., Bušta, J., Horák, A.: Expanding translation memories: Proposal and evaluation of several methods. RASLAN 2013 Recent Advances in Slavonic Natural Language Processing (2013) 71
2. Steinberger, R., Eisele, A., Klocek, S., Pilos, S., Schlüter, P.: Dgt-tm: A freely available translation memory in 22 languages. arXiv preprint arXiv:1309.5226 (2013)
3. Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., et al.: Moses: Open source toolkit for statistical machine translation. In: Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions, Association for Computational Linguistics (2007) 177–180
4. Gao, Q., Vogel, S.: Parallel implementations of word alignment tool. In: Software Engineering, Testing, and Quality Assurance for Natural Language Processing, Association for Computational Linguistics (2008) 49–57
5. Och, F.J., Ney, H.: A systematic comparison of various statistical alignment models. Computational linguistics **29**(1) (2003) 19–51
6. Heafield, K.: Kenlm: Faster and smaller language model queries. In: Proceedings of the Sixth Workshop on Statistical Machine Translation, Association for Computational Linguistics (2011) 187–197
7. Jakubíček, M., Kilgarriff, A., Kovář, V., Rychlý, P., Suchomel, V., et al.: The tenten corpus family. In: Proc. Int. Conf. on Corpus Linguistics. (2013) `http://www.sketchengine.co.uk/documentation/wiki/Corpora/enTenTen`
8. Denkowski, M., Lavie, A.: Meteor universal: Language specific translation evaluation for any target language. In: Proceedings of the EACL 2014 Workshop on Statistical Machine Translation. (2014)