Chapter 6

Introduction to Translation Memories

Translation Memories (TMs) are "structured archives of past translations" which store pairs of corresponding text segments[1] in source and target languages known as "translation units" (Simard 2020). TMs are used during the translation process in order to reuse previously translated segments. The original idea of TMs was proposed more than forty years ago when Arthern (1979) noticed that the translators working for the European Commission were wasting valuable time by re-translating (parts of) texts that had already been translated before. He proposed the creation of a computerised storage of source and target texts which could easily improve the translators' performance. This storage could be part of a computer-based terminology system. Based on this idea, many commercial TM systems appeared on the market in the early 1990s (Bowker 2006). Since then, the use of this particular technology has kept growing, and recent studies show that it is used on a regular basis by a large proportion of translators (Zaretskaya et al. 2018).

TM systems help translators by continuously providing them with so-called matches, which are translation proposals retrieved from its database. These

[1] Segments are typically sentences, but there are implementations which consider longer or shorter units.

matches are identified automatically by comparing the segment that must be translated with all the segments stored in the database. There are three kinds of matches: exact, fuzzy and no matches. Exact matches are found if the segment to be translated is identical to one stored in the TM. Fuzzy matches are used in cases where it is possible to identify a similar segment to the one to be translated. Therefore, it is assumed that the translator will spend less time editing the translation retrieved from the database than translating the segment from scratch. No matches occur when it is impossible to identify a fuzzy match (i.e. there is no segment similar enough to the one to be translated to be worth using its translation).

TMs distinguish between fuzzy and no matches by calculating the similarity between segments using a similarity measure and comparing it to a threshold. Most of the existing TM systems rely on a variant of the edit distance as the similarity measure and consider a fuzzy match when the edit distance score is between 70% and 95%.[2] The main justification for using this measure is the fact that the edit distance between two texts can be easily calculated, is fast and is largely language-independent. However, edit distance cannot capture the similarity between segments correctly when different wording and syntactic structures are used to express the same idea. As a result, even if the TM contains a semantically similar segment, the retrieval algorithm will not identify it in most cases. To make this clearer, consider the following three sentences.

[2]It is unclear the origin for these values, but translators widely use them. Most of the tools allow translators to customise the value of this threshold according to their needs. Translators use their experience to decide which value for the threshold is appropriate for a given text.

1. I like Madrid which is such an attractive and exciting place.

2. I dislike Madrid which is such an unattractive and unexciting place.

3. I love Madrid as the city is full of attractions and excitements.

Assume sentences 2 and 3 already had their translations in the TM database, and now sentence 1 has to be translated. The majority of the commercial TM systems based on edit distance return sentence 2 as a fuzzy match to the incoming sentence since the edit distance between sentences 1 and 2 are lower than sentences 1 and 3. However, sentence 3 is semantically closer to sentence 1 than sentence 2 and does not need many edits in the post-editing process. This nature of the edit distance based TM systems of not proving semantically close matches hinders the translators' efficiency (Ranasinghe et al. 2020a).

Researchers address this shortcoming of the edit distance metric by employing similarity metrics to identify semantically similar segments even when they are different at the token level. Section 6.1 discusses some of the approaches proposed so far. These approaches incorporate simple operations like paraphrasing to the TM matching process to provide semantically similar matches. As we observed in Part I of the thesis, deep learning based architectures are state-of-the-art in STS. Therefore, in Part II of the thesis, we propose a novel TM matching and retrieval method based on deep learning that can capture semantically similar segments in TMs better than the methods based on edit distance. As we discussed in Part I of the thesis, in addition to providing state-of-the-art results, deep learning based STS methods can easily be adapted in other

languages and domains, which is beneficial to TMs as they are employed in a wide range of domains and languages.

Utilising deep learning in TM matching methods bring obvious challenges regarding efficiency and storage. In Chapter 7, we discuss these challenges and carefully pick STS methods that can be efficient in the TM matching process. We evaluate these methods on a real-world TM, comparing them with the edit distance. As far as we know, this is the first study done on employing deep learning based STS methods in TM matching and retrieval. The main contributions of this part of the thesis are,

1. We perform a rigorous analysis on existing TM matching algorithms and identify the main shortcomings in them.

2. We propose a novel TM matching and retrieval algorithm based on deep learning and evaluate it on a real-world TM using English-Spanish pairs.

3. We compare the results of the proposed method with an existing TM system and show that our approach improves the TM matching and retrieving process.

The remainder of this chapter is structured as follows. Section 6.1 discusses the various TM matching algorithms and their shortcomings. In Section 6.2, we introduce the real-word TM we used for the experiments in this part of the thesis. Section 6.3 shows the evaluation metrics that we used to evaluate the experiments. The chapter finishes with the conclusions.

6.1 Related Work

As discussed before, even though TM systems have revolutionised the translation industry, these tools are far from being perfect. A serious shortcoming is that most commercial TM systems' (fuzzy) matching algorithm is based on edit distance, and no language processing is employed. Among the first ones to discuss the shortcomings were Macklovitch and Russell (2000) who showed that Translation Memory technology was limited by the rudimentary techniques employed for approximate matching. They comment that unless a TM system can perform morphological analysis, it will have difficulty recognising similar segments in the matching process.

The above shortcomings paved the way for developing second-generation TM tools, which had some language processing capabilities such as grammatical pattern recognition and performed limited segmentation at the sub-sentence level. However, there are only a few commercially available second-generation TM systems such as *Similis* (Planas 2005), *Translation Intelligence* (Grönroos and Becks 2005) and Meta Morpho TM system, *Morphologic* (Hodász and Pohl 2005). *Similis* (Planas 2005) performs linguistic analysis to split sentences into syntactic chunks or syntagmas, making it easier for the system to retrieve matches. *Morphologic* uses lemmas and part-of-speech information to improve matching, especially for morphologically rich languages such as Hungarian (Hodász and Pohl 2005). Even though the second-generation TM tools solved some of the issues in first-generation TM tools, Mitkov and Corpas (2008) discuss that they

still can not provide strong matches in most of the cases. Mitkov and Corpas (2008) show that none of the second-generation TM systems would be capable of matching *Microsoft developed Windows XP* with *Windows XP was developed by Microsoft* or matching *The company bought shares* with *The company completed the acquisition of shares.*

To overcome this shortcoming, Pekar and Mitkov (2007) developed the so-called third-generation TM tools, which analyse the segments not only in terms of syntax but also in terms of semantics. Pekar and Mitkov (2007) perform linguistic processing over tree graphs (Szpektor et al. 2004; Knight and Graehl 2005) followed by lexicosyntactic normalisation. Then similarity between syntactic-semantic tree graphs is computed, and matches at the sub-sentence level are established using a similarity filter and a node distance filter. While this promising work was the first example of matching algorithms for future third-generation TM systems, the described approach was not deemed suitable for practical applications due to its very long processing time (it could take days to compare matches). Another method that performs matching at the level of syntactic trees is proposed by Vanallemeersch and Vandeghinste (2014). The results presented in their paper are preliminary, and the authors notice that the tree matching method is "prohibitively slow".

Further work towards the development of third-generation TM systems included paraphrasing and clause splitting. Raisa Timonera and Mitkov (2015) experimented with clause splitting and paraphrasing, seeking to establish whether these NLP tasks can improve the performance of TM systems in

terms of matching. Furthermore, into this, Gupta et al. (2016b) experimented with incorporating paraphrasing to the TM matching algorithm to secure more matches. The authors sought to embed information from PPDB[3], a database of paraphrases (Ganitkevitch et al. 2013), in the edit distance metric by employing dynamic programming (DP) (Gupta et al. 2016b) as well as dynamic programming and greedy approximation (DPGA) (Gupta et al. 2016a). In more recent work, Gupta et al. (2014a) developed a machine learning approach for semantic similarity and textual entailment based on features extracted using typed dependencies, paraphrasing, machine translation, evaluation metrics, quality estimation metrics and corpus pattern analysis. This similarity method was experimented with to retrieve the most similar segments from a translation memory. But the evaluation results showed that the approach was too slow to be used in a real-world scenario (Gupta et al. 2014b).

With this analysis, we identified two key limitations in current third-generation TM systems. First, most of them rely on external knowledge bases, including WordNet and PPDB, which are challenging to use in many languages and domains. Secondly, the majority of these approaches are slow to be used in real-world applications. To address these limitations, we propose to use deep learning based STS metrics we experimented in Part I of the thesis in TM matching. As aforementioned, these methods do not depend on external knowledge bases, and most of them are optimised to use effectively in real-world scenarios. Therefore, in Chapter 7 we evaluate these STS metrics in TM matching

[3]PPDP is available on http://paraphrase.org/#/download

and retrieval. To the best of our knowledge, this is the first study to employ deep learning in translation memories.

6.2 Dataset

For the experiments of this part of the thesis, we used the DGT-Translation Memory[4] which has been made publicly available by the European Commission's (EC) Directorate General for Translation (DGT) and the EC's Joint Research Centre. DGT-TM contains official legal acts. It consists of sentences and their professional translations covering twenty-two official European Union (EU) languages and their 231 language pair combinations. The translations are produced by highly qualified human translators specialised in specific subject domains. It is typically used by translation professionals in combination with TM software to improve the speed and consistency of their translations. We should note that the DGT TM is a valuable resource for translation studies and for language technology applications, including statistical machine translation, terminology extraction, named entity recognition, multilingual classification and clustering, among others (Aker et al. 2013; Besacier and Schwartz 2015).

While we chose English-Spanish sentence pairs for the experiments of this study, our approach is easily extendable to any language pair. In this study, 2018 Volume 1 was used as the experimental translation memory and 2018 Volume 3 as input sentences. The translation memory we built from 2018 Volume 1 featured 230,000 sentence pairs whilst 2018 Volume 3 had 66,500 sentence pairs which we

[4]DGT-TM is available to download at https://ec.europa.eu/jrc/en/language-technologies/dgt-translation-memory.

used as input sentences.

6.3 Evaluation

TM systems are typically evaluated by measuring the *quality* of the retrieved segments from the matching algorithm (Gupta et al. 2015b). This *quality* is often considered to be the correspondence between the retrieved segment and the reference translation: *"the closer a retrieved segment is to a reference translation, the better it is"*. First, the quality scores are calculated for individual segments by comparing them with the relevant reference translations. These scores are then averaged over the whole corpus to estimate the quality of the TM system. Such quality evaluation techniques between the retrieved segment and the reference translation are called automatic metrics for machine translation evaluation.

Over the years, researchers have produced many automatic metrics for MT evaluations. BLEU (bilingual evaluation understudy) (Papineni et al. 2002) is the most popular and oldest automatic metric. BLEU was one of the first metrics to claim a high correlation with human judgements of quality and remains one of the most inexpensive metrics (Gupta et al. 2015a). However, using BLEU has drawbacks. The main drawback in BLEU is it does not consider the meaning and does not directly consider sentence structure (Sellam et al. 2020). Since this study aims to provide TM matches that are closer in meaning, we did not consider BLEU as our evaluation metric.

METEOR is a more recent automatic metric for MT evaluation that was designed to explicitly address several observed weaknesses in BLEU (Banerjee

and Lavie 2005). Similar to BLEU, METEOR is also based on explicit word-to-word matching. However, unlike BLEU, it not only supports matching between identical words in the two strings compared, but can also match words that are simple morphological variants of each other (i.e. they have an identical stem), and words that are synonyms of each other. Considering these advantages in using METEOR, we employed METEOR as our evaluation metric for the experiments in this part of the thesis.

It should be noted that the automatic evaluation metrics are far from being perfect (Sellam et al. 2020). These metrics have their own limitations, which can affect the evaluations of this study. Whatever the automatic evaluation metric we use, we would not be able to avoid these weaknesses completely. Therefore, in addition to the automatic evaluation, we carried out a human evaluation. We asked three native Spanish speakers with a background in translation studies to compare the segments retrieved from our algorithm. In Chapter 7, we report these results alongside the automatic evaluation metrics.

6.4 Conclusions

The Translation Memory (TM) tools revolutionised the work of professional translators, and the last three decades have seen dramatic changes in the translation workflow. One of the essential functions of TM systems is their ability to match a sentence to be translated against the database. However, most of the current commercial TM systems rely on edit distance to provide TM matches. Despite being simple, edit distance is unable to capture the similarity between

segments. As a result, even if the TM contains a semantically similar segment, the retrieval algorithm will not be able to identify it. This can hinder the performance of translators who are using the TM.

As a solution to these limitations, the second-generation and third-generation TM systems are proposed. However, they are far from being perfect. Most of them lack the efficiency which is required for TM systems. Furthermore, they rely on language-specific knowledge bases, which makes them less adaptable to other languages and domains. Therefore, to overcome these shortcomings, we propose a novel TM matching and retrieval algorithm based on STS methods we experimented with in Part I of the thesis. In addition to providing state-of-the-art STS results, these algorithms are fast and easily adaptable to other languages and domains, which is beneficial for TMs.

We will be using English-Spanish sentence pairs in DGT translation memory as the dataset for our experiments. Our evaluation will be based on METEOR, an automatic metric for MT evaluation. Furthermore, considering the limitations in automatic metrics, we would also incorporate a human evaluation in our experiments. The proposed method, results and evaluation will be explained in detail in Chapter 7.