**UNIVERSITY OF ZAGREB**
**FACULTY OF HUMANITIES AND SOCIAL SCIENCES**
**DEPARTMENT OF ENGLISH**

GRADUATE PROGRAMME
TRANSLATION TRACK

**Margita Šoštarić**

# Advanced fuzzy matching in the translation of EU texts

Diploma thesis
Research paper presented in fulfilment of requirements for a second-cycle degree

Supervisors:
Nataša Pavlović
Ivana Simeon

2018

**Acknowledgements**

"Nothing ever goes according to plan. The key is to stop planning."

V. V.

"The difference between theory and practice is much smaller in theory than in practice."

T. V.

A great heartfelt thanks to the employees of KU Leuven's Centre for Computational Linguistics, who not only generously decided to open their door (and servers) to me and give me a chance to do this research, but also made an effort to make Belgium feel like a second home to me during my stay. I especially need to thank Vincent Vandeghinste and Tom Vanallemeersch for all their help and guidance, but most of all for their relentless patience throughout my six-month internship. As an utter beginner in the world of computational linguistics, I could have hardly wished for a more supportive working environment. Thanks are also due to my dear colleagues and friends, Milan, Bram B. and Bram V., for being a most pleasant company at the office and even putting up with me outside it. Cheers, guys, thank you for not letting me work too much, for being my shoulders to rant on and for tirelessly trying to refine my (still hopeless) taste in beer.

Back in Croatia, I need to thank my two supervisors, Nataša Pavlović and Ivana Simeon, for agreeing to jointly take on the task of supervising this thesis. I am especially grateful to professor Pavlović for supporting and encouraging my wish to pursue this somewhat untypical research topic, even though we knew from the start that it would pose quite a challenge for both of us. Finally, I must not forget to thank my wonderful family and friends for all their love and support, and for somehow always managing to find the right words… even though to this day none of them actually get what my thesis is about.

**Table of contents**

**Abstract**

In the translation industry today, CAT tool environments are an indispensable part of the translator's workflow. Translation memory systems constitute one of the most important features contained in these tools and the question of how to best use them to make the translation process faster and more efficient legitimately arises. This research aims to examine whether there are more efficient methods of retrieving potentially useful translation suggestions than the ones currently used in TM systems. We are especially interested in investigating whether more sophisticated algorithms and the inclusion of linguistic features in the matching process lead to significant improvement in quality of the retrieved matches. The used dataset, the DGT-TM, is pre-processed and parsed, and a number of matching configurations are applied to the data structures contained in the produced parse trees. We also try to improve the matching by combining the individual metrics using a regression algorithm. The retrieved matches are then evaluated by means of automatic evaluation, based on correlations and mean scores, and human evaluation, based on correlations of the derived ranks and scores. Ultimately, the goal is to determine whether the implementation of some of these fuzzy matching metrics should be considered in the framework of the commercial CAT tools to improve the translation process.

**Key words**

translation memories, CAT tools, fuzzy matching, similarity metrics

## 1. Introduction

Ever since they entered into usage in the mid 1980s (Seal, 1992), the importance of translation memory systems in the translation process has steadily grown. Today, they are an indispensable technology in the translator's workflow (Lagoudaki, 2009; Simard and Fujita, 2012). Despite their widespread usage, as well as the fact that they have been present in the translation industry and successfully integrated into CAT tool environments for a reasonably long time, relatively little improvement has been made in their core functioning (Simard and Fujita, 2012; Reinke, 2013). They have recently attracted a lot of attention from research communities and a large number of papers were written on different topics related to TMs, from surveying translators' opinions on and expectations from TM systems (Lagoudaki, 2009; Moorkens and O'Brien, 2016; Federico et al., 2012; Parra Escartín, 2015), to evaluating and improving similarity metrics for searching the memories (Hodász and Pohl, 2005; Pekar and Mitkov, 2007; Baldwin, 2010; Bloodgood and Strauss, 2014; Simard and Fujita, 2012; Gupta et al., 2014b; Vanallemeersch and Vandeghinste, 2015a; Gupta et al., 2016), to the ethical aspects and copyright issues concerning storing and reusing both the original source text and its translated counterpart (Pym, 2003; Blésius, 2003; Drugan and Babych, 2010). Moreover, the usefulness of TMs as high-quality, human-produced parallel corpora had been overlooked as a valuable resource for improving machine translation and it is only relatively recently that the MT researchers recognised their benefits and started using them in developing MT systems (Simard and Fujita, 2012). These two translation technologies are strongly linked in both theory and practice, and most available CAT tools already offer some, computationally more or less sophisticated, possibility of combining the use of TM and MT systems in their environments (Lagoudaki, 2009; Reinke, 2013).

However, despite ample research, a number of problems pertaining to TM systems and CAT tools on the whole remain unsolved in the commercial products used in the translation industry. The aim of this research is to examine the possible improvement of one such issue – the similarity metrics used for searching through the translation memories and retrieving the relevant matches to be offered to translators as translation suggestions for a particular segment. Although it has never been publicly disclosed, it is widely believed that most CAT tools use some variant of edit distance (Bloodgood and Strauss, 2014; Simard and Fujita, 2012; Koehn and Senellart, 2010; Christensen and Schjoldager, 2010; He et al., 2010), a fairly simple, but relatively efficient similarity metric often used for a variety of data comparison purposes. However, the metric's limitations become painstakingly obvious when it is applied

to morphologically rich languages, as it has problems dealing with inflectional phenomena and, at least in its basic implementation, does not allow for changes in word order. At a more profound level, since the metric only searches for exact overlap in sequences of word forms, it cannot account for semantic similarity of the segments as perceived by human translators (Gupta et al., 2014b). Consequently, although the metric performs very well on highly similar sentences, it cannot cope with segments whose similarity lies in aspects which are less straightforward than exact words.

In this research, the described rudimentary implementation of edit distance is used as the baseline against which the performance of a number of different similarity metrics is tested. The aim is to examine whether the shortcomings of edit distance can be overcome by using different similarity algorithms and including different levels of linguistic knowledge in the matching process. The potential improvements in the matching process should lead to more useful translation suggestions being offered to translators, hence speeding up the translation process. To measure this "usefulness", the matches retrieved by the similarity metrics are automatically evaluated using evaluation metrics, but a number of them were also given to human evaluators in the form of a survey[1], since the quality of the performance of metrics should primarily be estimated by the end users of TM systems. Should some of the metrics prove to perform better than the widely used edit distance, their implementation in the CAT tools should ultimately be considered in order to improve matching and hence further enhance and speed up the translation process[2].

## 2. Theoretical background and related research

As already mentioned, the research on translation memory systems has been done from various perspectives and with various aims in mind. In this section we give an overview of a number of papers dealing with topics closely related to our own. On the other hand, we put the aims of our research and the used approach into perspective by discussing a number of concepts and notions within the wider theoretical context of translation studies.

### 2.1. Introduction to translation memory systems

According to Reinke (2013), the importance of translation memories has become especially prominent in the context of the rapidly expanding translation market, making TMs "the major language technology" in the translation industry (Reinke, 2013: 27). Translation memories are

[1] We would like to hereby thank the translators who took the survey for their efforts and useful feedback.
[2] This research was done within the framework developed for the purposes of the SCATE project, carried out at KU Leuven's Centre for Computational Linguistics. Its methodology was largely based on the approach used in the paper *Assessing linguistically aware fuzzy matching in translation memories* (2015) by Tom Vanallemeersch and Vincent Vandeghinste.

aligned parallel corpora comprised of translation units (TU), that is, source segments coupled with their translations. In other words, TMs are databases containing previously translated texts, fragmented and aligned at sentence level[3], so that they can be searched, edited and, ultimately, reused in future translation tasks (Sikes, 2007). When the translator is translating a new text, a similarity algorithm searches through the TM and retrieves from it the segments whose similarity with the currently translated segment is estimated at a value above some required threshold[4]. These matches can be identical to the translated segment (exact matches) or display a certain degree of similarity (fuzzy matches), based on which the translator chooses whether to accept them as translation suggestions and post-edit them, or discard them and translate from scratch. The possibility of referring to verified existing translations has proved extremely useful in localization industry and the translation of specialised texts, as it significantly increases the speed of the process due to the repetitive character of these texts (Lagoudaki, 2009; Reinke, 2013; Christensen and Schjoldager, 2010). According to recent reports, most translators consider TM systems useful and gladly rely on them in translation (Moorkens and O'Brien, 2016; Zhechev and van Genabith, 2010). Moreover, with the growing demand for quick translation of large amounts of text changing the make-up of the translation process itself, the workload often exceeds the capacities of a single translator and calls for well-coordinated translation projects. In this case, TMs are useful for ensuring consistency among the translators and help project managers partition the text more logically when distributing the workload (Parra Escartín, 2015).

In this research, we focus on another domain which has greatly profited from the advent of TM systems – legal translation in the context of the European Union. Due to its particular nature and norms, legal translation has recently attracted a lot of attention from different research communities, with a number of terminological and lexical resources and tools being developed to facilitate the translation process and increase its consistency (Biel and Engberg, 2013). Within this domain, the translation of the EU legislative texts constitutes a marked phenomenon, due to the unprecedented multilinguality of its context (Felici, 2010). The great number of languages and large amounts of text have made TM systems particularly important in the translation of EU documents, and it is around these texts that we focus our research. As legal language typically displays a high degree of formulaicity and standardization in

[3] It is in principle so that TUs consist of a source and target sentence and in literature are therefore often referred to as sentence pairs. However, as the granularity of aligned text can vary from smaller units (e.g. headings, table or list contents) to larger chunks of text (e.g. the source sentence is split up into two sentences on the target side), it is more accurate to use the more general term *segment* instead of *sentence* when speaking of TUs.

[4] The general practice in the translation industry seems to be setting the fuzzy match threshold at 70 percent.

terminology and structure (Biel and Engberg, 2013), the decision to use this particular dataset will inevitably have some implications for the expected research outcomes. Most notably, although the lexical aspect of texts unquestionably carries significant weight in all types of translation (Simard and Fujita, 2012), in legal translation it becomes even more pronounced due to the strictness of expression. The possible effects of this general restriction in language variability are further discussed later in the context of fuzzy matching.

Finally, although the value of TM systems in the translation industry is indisputable, we must address an issue implicitly built into the very core of their functioning, and that is the problematic nature of text segmentation and the hazardous effect it might have on the integrity of the translation (Pym, 2006). As sentences in Indo-European languages generally do contain well-rounded grammatical units and express "complete thoughts" (Timonera and Mitkov, 2015: 17), splitting the text at sentence level for the purposes of translation seems justified. However, a lot of information is also contained in the surrounding text and TM systems are currently unable to utilise the stylistic, discursive and contextual information in a suitable way to improve their performance and overall translation quality.

### 2.2. Translation suggestion usefulness – fuzzy matching metrics

One of the fundamental features of TM systems are the matching algorithms used to retrieve translation suggestions from the TMs. Similarity algorithms generally have a broad scope of application and a great number of them have been developed for different purposes and in different scientific disciplines. In this subsection, we introduce a number of similarity algorithms which can be used in the context of fuzzy matching, and discuss some of their advantages and drawbacks.

Despite the immense variety in ways of establishing similarity between two compared segments, commercial TM systems persist with using some variation of edit distance, such as Levenshtein distance (Levenshtein, 1966). In its basic form, this metric is a simple equally-weighted edit distance, which means that it only counts the number of editing operations, substitutions, insertions and deletions, performed on words to turn one string into another[5]. An obvious drawback to this metric is that it does not allow for word crossings, i.e. it calculates the minimum edit distance matrix given a fixed word order. One way of overcoming this problem is using bag-of-words metrics, such as percent match (cf. Bloodgood and Strauss, 2014; Baldwin, 2010), which count the shared elements regardless of

[5] Alternatively, the operations can be assigned different costs. Substitution is then usually "costlier", as it involves both deletion and insertion (Jurafsky and Martin, 2009). The most frequently used implementations are run at word or character level.

their position in the segments. The problem with this metric is that it counts each appearance of a particular word as overlap. This usually results in assigning too much weight to highly frequent words, such as function words, whereas their usefulness in a translation situation is generally of limited extent[6]. One possible way of dealing with this problem would be to use linguistic information and give more weight to certain parts of speech or generally content words. Another approach would be to combine the metric with IDF weights[7] (Bloodgood and Strauss, 2014) to tone down the importance of often recurring function words in matching.

Generally speaking, there are endless possibilities in constructing different weighting schemes, which can be integrated with the matching algorithms to effectively give prominence to certain desired features, making the metrics less coarse and absolute in handling the complexity of language phenomena. However, weights do not solve the inherent risk of bag-of-words approaches: placing focus on single elements can potentially excessively fragment the text and fail to reflect the dependency relations implicitly contained in word order. As the translator would presumably want the fuzzy match to share more than a number of sporadic words with the sentence he or she is translating, it might be desirable to somehow match larger phrasal structures. Hence, the assumption behind ngram-based approaches[8] is that higher-order ngrams are preferred over shorter spans, as they constitute more meaningful overlapping units and preserve the grammaticality within the matched phrases. In their study, Bloodgood and Strauss (2014) tested the usefulness of preserving the local context in fuzzy matches by creating a weighted version of ngram precision in which translators could themselves set the preferred length of matching spans. Human evaluators judged this metric to work very well when shorter ngrams were allowed to contribute more to the final match score, as these matches retained a degree of coherence without sacrificing too much of the variability and flexibility in matching.

Regardless of their differences, all of these three approaches use language independent metrics run on surface form of the tokens contained in the compared segments. Moreover, the pointed out problems and solutions were generally discussed with respect to languages which are less morphologically diverse and have a stricter word order. Considering the fact that TM systems should perform well for a variety of different languages, the metrics described above

[6] A typical illustration of this is the frequently recurring articles. For example, if the article *the* appears once in a short query segment and multiple times in the matching segment, this might suffice to estimate it as a good match, although in reality it is highly unlikely that it constitutes a useful translation suggestion.

[7] Inverse document frequency (IDF) is calculated across the corpus and assigns weights to words based on frequency of their occurrence. The underlying assumption is that translating the words with lower frequency will be more valuable and they are hence given more prominence in the matching process.

[8] In our research framework, ngrams are word units, i.e. unigrams denote single words, and higher-order ngrams are units consisting of multiple (*n*) words.

might lack the flexibility needed to account for phenomena in highly flectional languages. One simple way of dealing with this would be to run the matching process on units below word-level, for instance on single characters or shorter sequences of characters. Alternatively, we can try to improve the performance of the metrics by including linguistic features in the matching process. For instance, the same string-based metrics can be applied to different matching items, containing various types of linguistic information, such as stems or lemmas, part-of-speech tags, dependency structures or head-word chains (cf. Vanallemeersch and Vandeghinste, 2015a). Taking this idea of linguistically aware matching further, we can opt for metrics which do not compare strings, but tree structures and the data contained in them. There is a large variety of those metrics as well, such as tree edit distance (Klein, 1998) and various metrics drawing on the information contained in tree and subtree alignment (Jiang et al. 1995; Liu and Gildea, 2005; Zhechev and van Genabith, 2010; Vanallemeersch and Vandeghinste 2015a). However, what should be kept in mind is that using tree-based metrics is inevitably far more complex and computationally heavy than measuring string-based similarity, with the very generation and storage of parse tree structures already being expensive in terms of time and memory. Therefore, attempts have been made to "flatten" the complex parse trees into the more easily manageable string form, while retaining all information contained in the nodes. One of the approaches proposes using Prüfer sequences (Prüfer, 1918) to convert the information contained in trees into a string form (Li et al., 2008) and assigning different values to the different types of overlapping items when comparing two such segments (Vanallemeersch and Vandeghinste, 2015a).

Apart from syntactic linguistic features, similarity metrics can also be applied to semantic information. For instance, in order to improve the recall and provide translators with better and lexically more diverse fuzzy matches, Gupta et al. (2016) propose using a large paraphrase database alongside the basic edit-distance algorithm. In their approach, they create an additional TM augmented with paraphrased structures. Both TMs are used in matching, with the matches from the original TM generally given advantage to in score ties. This seems like a viable framework, since they do not make the matching algorithm itself more complex and the matching remains fast. The paraphrase tables are relatively easy to develop as a resource from parallel corpora, but more linguistic features would be required for a successful integration of paraphrases when working with highly flectional languages and, as is always the case with "general" linguistic resources, these paraphrases should ultimately be somehow constrained with regard to their adequacy in particular domains and contexts. Several other, more complex matching algorithms aimed at semantic similarity are discussed in the

following subsection on evaluation metrics. We should also mention that there have been numerous attempts at intelligently combining multiple metrics (Gupta et al., 2014b; Bär et al., 2012, Vanallemeersch and Vandeghinste, 2015a), in order to maximise their strengths and smooth out their faults.

In another recent approach, Timonera and Mitkov (2015) propose chunking the TM segments into phrases or clauses and doing sub-segment matching. Although this approach significantly increases the recall, the question of how useful that is in practice actually points to two inherent limitations of TM systems. First, fragmenting a text into even smaller units than the default sentence level highlights that these systems remain primarily intended for the translation of highly repetitive specialised texts (Reinke, 2013). Second, it brings out the somewhat paradoxical nature of the fuzzy matching task itself: its goal is to provide the translator with a segment in the target language based on a comparison done on the source side. As particular expressions, norms and structural features can vary to different extents between languages, the sub-segment similarity on the source side need not be present to the same extent on the target side[9], emphasising the fact that TM systems work better for structurally similar language pairs (Parra Escartín, 2015). To address the latter issue, the matching process should ideally be constrained by taking into account the target side of the TU (cf. Ma et al., 2011). As for the former, it certainly stands to reason to include linguistic features and resources in matching instead of restricting the text and its complex language phenomena to surface forms. However, to what extent we use these features is not a straightforward question, as translation is relatively bound by concrete lexical choices in the source text. Moreover, the usefulness of the translation suggestion also depends on a number of external factors, such as the purpose of the text, the type of text, the relevant norms, the target audience and a whole range of other phenomena influencing the decisions made by the translator in particular contexts. These notions have been extensively discussed within a number of theoretical frameworks in translation studies, but as the current CAT tools can take these circumstances into account only to a very limited extent[10], we will not elaborate on them further. They are, however, taken at least partially into account in the form of human

[9] To illustrate this, we can give the example of compounds in Germanic languages or flectional endings in Slavic languages. A high-scoring match in analytic languages such as English can hence still elicit a substantial amout of post-editing on the target side, or even render the offered match unusable.

[10] For instance, working with the translators from the European Parliament, we have been told that their translation process is "contextualised" in terms of inter-textual references by instructing the system to prioritise certain relevant reference documents over others when searching through the TMs. This is a simple way to indirectly ensure the consistency important in this particular type of translation. We are not familiar with any other features which would enable the situational context to influence the matching process being implemented in the current CAT tools.

judgment of the metrics' performance, with the subjective factor of individual translators' preferences further complicating the image.

### 2.3. Translation quality – automatic evaluation metrics

Unlike in research on machine translation, the question of *quality* is not as central when dealing with translation memories, since the output of TM systems is human-produced translation and therefore should, at least nominally, unquestionably be of high quality. The reason why we give an overview of the metrics aimed at estimating translation quality is twofold: first, as they are used to evaluate the performance of the matching metrics, we consider it important to discuss the underlying assumptions about translation quality these metrics are built on; second, as noted by Simard and Fujita (2012), with only slight adaptations made to the algorithms, metrics for automatic evaluation of MT can themselves be used as fuzzy matching metrics in TM systems[11]. Unfortunately, the question of how to define translation quality is in no respect a straightforward issue. As Koby et al. (2014) jokingly note, it is already impossible to strictly define *translation* and *quality*, let alone formulate an absolute definition of *translation quality*. We might add that it is one thing to broadly define it through contemplative discussion in theory[12], and entirely another to find a plausible way of formalising and quantifying it in practice. As this matter is crucial for the research and system development in the field of translation technology, considerable effort has gone into constructing a reliable automatic evaluation framework which would reflect the vague idea of *quality* as perceived by humans. To be able to handle the broad scope of the notion in some formal way, the evaluation metrics have inevitably had to reduce it to a certain aspect (Banarjee and Lavie, 2005), their features then being attuned to (more or less successfully) capturing particular ways in which these phenomena are supposedly reflected in text[13].

Although often vigorously disputed, BLEU (Papineni et al., 2002) is currently the most frequently used automatic evaluation metric (Lo and Wu, 2011). This metric constitutes a fairly flat, though computationally efficient method of comparing segments based on lexical similarity through a combination of ngram precision and a brevity penalty. To put it simply, BLEU compares the two segments by measuring the proportion of the matching ngrams to the total number of ngrams in the evaluated segment. As it does not directly take into account this

[11] To distinguish between the algorithms used as fuzzy matching metrics and as evaluation metrics, the subscript "T" is added when referring to the latter, denoting the *target side*.

[12] House (2000) notes that for an assessment of translation quality, you need a translation theory, emphasizing that there are no absolute parameters for the estimation of quality.

[13] The aspects of quality that we discuss in this section (fluency, adequacy, accuracy) are used in the sense as defined in the framework of White et al. (1993).

same proportion in the reference segment, BLEU uses the brevity penalty to account for this lack of recall, i.e. it assigns penalties for differences in length between the compared segments. The quality measured by BLEU is defined in terms of fluency, represented indirectly by overlaps in higher-order ngram spans, and adequacy, reflected in shorter ngram overlaps. Leaving aside the question of BLEU's efficiency in measuring the phenomena it purports to capture, comparing segments at this flat level effectively requires a diversity of references that a query is compared to, in order to account for the possible lexical variation. A step towards resolving this was taken by METEOR (Banarjee and Lavie, 2005). METEOR is an edit-distance-based metric which measures similarity on surface lexical forms and their stemmed versions, but also makes provision for the fact that the same meaning could be expressed in various ways by incorporating models for identifying synonyms (usually built from WordNet[14] or similar resources) and paraphrases in the compared segments. The modules are weighted in the final score calculation, the assumption being that having an exact match is better than having a synonymous or paraphrased alternative[15]. Although METEOR proved to correlate much better with human judgment (Denkowski and Lavie, 2014), an obvious drawback is its inapplicability to under-resourced languages, as well as relative inflexibility in handling variation in word order because of the penalties assigned to word crossings. There are a number of other edit-distance-based evaluation metrics, aimed at estimating quality in terms of adequacy by measuring the cost of edits, or "error rate", such as WER, PER and TER[16] (Snover et al., 2006). TER reportedly correlates rather well with human judgment of quality (Lo and Wu, 2011), and its basic principle can be paraphrased as an equally-weighted count of all the edits made to convert a query segment into a reference segment (insertion, substitution, deletion, phrasal shifts, changes in punctuation and miscapitalisation), normalised across the length of the reference. Most automatic evaluation metrics also have human-targeted variants, which utilise manual annotation or pooled human feedback on translation quality to more profoundly tune the metrics' parameters.

Much like with the fuzzy matching metrics, there have been attempts to move beyond the lexical level and measure similarity on more abstract linguistic units, using for instance syntactic trees or semantic roles. Approaches aimed at semantic similarity can include shallow

[14] https://wordnet.princeton.edu/

[15] This makes sense if we consider the fact that not all synonyms are mutually exchangeable in all contexts. To take an example from the article, the words *computer* and *workstation* will be marked as overlap, but given a score of only 0.3 (Denkowski and Lavie, 2005). However, METEOR's statistical approach does not make it fully capable of dealing with phenomena pertaining to the register and stylistic features of a text.

[16] Although its original name is Translation Edit Rate, *edit* is frequently exchanged for *error* by analogy to word error rate (WER) and position-independent word error rate (PER). Either way, it is important to note that these metrics indicate higher similarity by lower scores, i.e. the fewer the errors/edits, the better the match.

semantic knowledge as a feature in aggregate metrics such as ULC (Giménez and Màrquez, 2007) or be entirely based on matching on semantic roles (Lo and Wu, 2011; Vanallemeersch and Vandeghinste, 2015b). According to the creators of the recently developed MEANT metric (Lo and Wu, 2011), the quality of translation essentially lies in the accuracy of the representation of the basic event structure. The metric thus transposes the notion of semantic similarity from the lexical level to semantic frames, which are used in comparisons. Although the paper reports positive results, the metric is not yet fully automated and its application requires resources and tools such as semantic parsers which are not available for a great number of languages. Needless to add, handling meaning from any perspective often poses problems which cannot be uniformly resolved, and modelling meaning through the strict computational framework is far from being a straightforward task. Whether the fact that the translation correctly conveys the essential relations of "who did what to whom" (Lo and Wu, 2011: 220) suffices to evaluate it as *good* could be disputed, but this approach presents an interesting broadening of the view on how to capture semantic similarity.

On the other hand, modelling syntactic knowledge seems to be, at least nominally, a slightly easier task and various resources and tools have been developed for a larger number of languages. Therefore, many researchers have tried to enhance their systems by incorporating syntactic information to improve the identification of shared constructions, grammaticality and word order variation (Liu and Gildea, 2005; Owczarzak et al., 2007). The idea of creating a weighted combination of multiple levels of similarity has also been explored in the sphere of automatic evaluation. For example, LAYERED (Gautam and Bhattacharyya, 2014) combines a lexical, syntactic and semantic layer, while BEER (Stanojević and Sima'an, 2014) is based on the permutation of tree nodes, but also takes into account the lexical layer. BEER draws on similar lexical resources as METEOR, but unlike the latter, it matches on more fine-grained character-based ngram orders and distinguishes between content and function words in matching. These different layers of linguistically aware and unaware features are combined using logistic regression.

However advanced these metrics may be, their limitations as valid estimators of quality in research similar to this one quickly come to light and their output should always be interpreted with a pinch of salt. For instance, studies have shown that metrics which capture a particular linguistic aspect, such as semantic similarity, correlate badly with evaluation metrics insusceptible to such features (Simard and Fujita, 2012). On the other hand, research has also shown that using similarly functioning algorithms both for matching and for evaluation results in self-selection bias, i.e. the evaluation metrics tend to correlate best with

essentially similar matching metrics and rate their performance higher (Simard and Fujita, 2012; Vanallemeersch and Vandeghinste, 2015a; Wolff et al., 2016). These two phenomena seem logical, but they are not always equally obvious. After all, the matching metrics on the source side and the evaluation metrics on the target side are ultimately applied to different language systems (Simard and Fujita, 2012). One way of obtaining more legitimate results is to use multiple evaluation metrics, with the tested setup ideally showing improvement according to all criteria.

Automatic evaluation is indeed indispensable in research and it is of unquestionable value when there is no other recourse. However, it is indisputable that TM and MT systems should ultimately be evaluated by the end users to obtain a more realistic image of the quality of their performance. As human evaluation is time-costly, noisy and expensive, it is inconvenient for large-scale evaluation tasks or for rough estimations of relative improvement at the development stage of the research, which frequently leads to the qualitative analysis and human judgment being entirely omitted from studies. On the other hand, how to approach and quantify the phenomena one wishes to measure constitutes an interesting and complex topic in its own right, the discussion of which lies beyond the scope of this research. Just as an illustration, at the next stage of this research, that is, should the implementation of particular metrics be considered, a more elaborate investigation of their performance could be conducted by measuring the gains in speed and reduction in post-editing effort. Approaches such as those proposed by Federico et al. (2012) or Green et al. (2013) would enable us to more directly quantify the actual usefulness of the retrieved matches in practice. However, for the purposes of our research, a much simpler task of ranking the offered matches should suffice to see how well the individual metrics correlate with the human judgment of usefulness. We present our approach in more detail in the section on methodology.

## 3. Aims and hypotheses

Before we proceed to describe the experimental setup, we give a short overview of the research aims and hypotheses, based on the above discussed notions and findings of previous research. In a general sense, the main aim of this research is to investigate whether there are more efficient ways of performing fuzzy matching than the surface-level edit-distance methods currently used in the translation industry. The hypothesis we build our experiment around is that algorithms which calculate similarity scores in a slightly more sophisticated way correlate better with the human judgment of usefulness. Another hypothesis is that augmenting the matching process with various types of linguistic information also leads to the

retrieval of more useful matches. Hence, algorithms of varying complexity are applied to the dataset in a number of configurations, to see whether some of them are more successful in capturing the notion of similarity as perceived by humans and in estimating the quality of matches as translation suggestions. The research primarily aims for improvement in the matching process in the lower scoring ranges, as the assumption is that some of these matches might still be useful, but the surface-level metrics fail to identify these aspects of similarity to the query segment. The only restriction to the expected outcomes of the research is imposed by the nature of the used dataset: as we are working with legislative EU texts, the added value of lexical and syntactic flexibility might be slightly less than in some other types of text, but these features are still expected to contribute to the improvement in the matching process. Ultimately, the practical purpose of the research is to examine whether an implementation of the tested matching algorithms should be considered in order to improve the translation workflow, should they prove to consistently perform significantly better than the metric currently used in the commercial CAT tools.

## 4. Methodology

The way in which we construct the experimental setup largely builds on the work done within the framework of the SCATE project[17], and the research itself was primarily envisioned as an extension of the research on fuzzy matching metrics presented in the paper by Vanallemeersch and Vandeghinste (2015a), aimed at examining whether using linguistic features leads to improvement in matching on the Europarl dataset (Koehn, 2005) for the language pair English-Dutch. Apart from the already discussed differences in the expected outcomes with regard to the used dataset, the used approaches differ in several other instances, the most important one being the inclusion of human evaluation.

### 4.1. Fuzzy matching metrics

As can be seen in the literature overview, there is a wide variety of similarity algorithms which can be used for fuzzy matching. In this subsection, we will give an overview of the basic functioning of the algorithms we opted for in this research[18].

#### 4.1.1. Individual fuzzy matching metrics

As already mentioned, the current commercial TM systems are believed to use some variation of edit distance as the similarity algorithm. If we are looking to investigate the possible improvements of these systems, it stands to reason to take this metric as our baseline

[17] https://www.arts.kuleuven.be/ling/ccl/projects/scate

[18] For brevity sake, the formulae of the matching algorithms can be found in Appendix I. In order to obtain comparable results, all metrics are normalised to give a score ranging from 0 (no overlap) to 1 (exact match).

in the research. We use a very basic implementation of Levenshtein distance: it compares the segments based on surface word forms, assigns each substitution, deletion or insertion the cost of 1 and does not allow for word-crossings when calculating the minimal distance matrix. The coarseness of this metric is visible from its very description, but the reason why it is still widely used in TM systems is presumably because it is fast and performs well on segments with a high percentage of similarity, which makes it a surprisingly strong baseline to test against. Other two surface-form metrics that we used are Percent match (PM) and Ngram precision (NGP) (Bloodgood and Strauss, 2014). The idea behind using them is to examine the usefulness of matches retrieved by a fairly rudimentary bag-of-words metric (PM) and a metric trying to take into account the local context of larger structures by identifying longer stretches of overlapping ngrams (NGP). The first one hence increases recall, whereas the second one aims for precision. PM very freely calculates the percentage of elements in the query found in the TM segment and is normalised only across the length of the query, which means that it can give a high score to a fuzzy match regardless of the matching segment length[19]. In contrast to that, NGP looks for overlapping sequences of elements of length 1 up to N, and its normalization enables us to control the segment length preference by changing the value of the Z parameter[20]. Aiming for higher precision, we decided to match on higher-order ngrams and a lower value for Z (set at 0.3). As the SCATE framework provides the possibility of running these three metrics on different elements of string-structured data, we also made linguistically informed variants of these metrics, applying them to lemma sequences, part-of-speech tags and Prüfer sequences. The parameters for running the metrics (such as weighting schemes and ngram lengths) were set experimentally by using the hill-climbing algorithm. Based on the number of varied parameters, the hill climber was applied with two or three random initializations to 10.000 segments from the training dataset, and the parameters were optimised for five best matches using mean $TER_T$ score[21]. The final configurations used in matching are presented in Appendix I[22].

[19] For instance, if the query is a phrase consisting of two words and both of those words (not even necesarrily constituting the same phrase) are found in a TM segment which is a long sentence, this metric will give this match a high score.

[20] By setting this parameter to a higher value, we effectively allow the algorithm to retrieve longer matches.

[21] Although it is arguably always better to determine optimal parameters experimentally rather than setting them intuitively and arbitrarily, there are a number of limitations to this approach which need to be kept in mind. First of all, the dataset the algorithm was run on is relatively small and consists of random segments extracted from the training set, so we can hardly claim its representativeness for the entire corpus. TER was also arbitrarily chosen as the optimization metric. Nevertheless, we estimated that this approach would suffice for the purposes of this stage of the research.

[22] We mark these metric variants with subscripts, e.g. $LEV_{LEM1DEF}$, $PM_{POS3DEF}$, $NGP_{PRUF4PRUF}$.

The rest of the metrics we applied were run with default settings. The first among them is another edit-distance metric, TER[23] (Snover et al., 2006). However, as the score it assigns is based on the cost of shifts and its value in theory has no upper boundary, to enable comparison with other matching metrics and mitigate the correlation to automatic evaluation, we use the inverted score, normalised to give back results ranging from 0 to 1 (Vanallemeersch and Vandeghinste, 2015a). Two more MT evaluation metrics are used as fuzzy matching metrics on the source side: METEOR (Banarjee and Lavie, 2005) and BEER (Stanojević and Sima'an, 2014)[24]. These metrics incorporate lexical semantic knowledge and, as mentioned before, BEER also makes use of syntactic information through node permutation. This leads us to the last metric we applied to the dataset – Shared Partial Subtrees (SPS), a metric which compares pairs of parse trees by identifying the overlapping subtree structures the trees share (Vanallemeersch and Vandeghinste, 2015a). The final score is calculated on the optimal combination of all shared subtrees, which are in turn individually scored based on the number of nodes, as well as on word relevance and the lexical and non-lexical similarity of the nodes.

#### 4.1.2. Combination of fuzzy matching metrics

As a final step, we decided to combine the above metrics to see if the matching can be improved by their combined impact. As mentioned in the literature overview, this idea is not new, as a number of different models have been developed both for the purposes of retrieval in TM systems and of evaluation in MT systems[25]. These models vary in complexity and the number of features, and in terms of that, our model is fairly simple and naïve. Its logic resembles that behind the log-linear model constructed by Bär et al. (2012) and regression trees constructed by Vanallemeersch and Vandeghinste (2015a), inasmuch as it uses pre-calculated scores by the metrics as feature values and takes the predicted value as the new fuzzy match score. After testing a number of setups, we opted for the Random Forest Regressor[26]. This is a very simple, but efficient ensemble learning method which uses a number of regression trees as the base algorithms and outputs their mean prediction. The fact that it averages out the result of multiple trees and that it trains different trees on different parts of the training dataset reduces the variance and makes the model less prone to

[23] The used version is 0.7.25 (see http://www.cs.umd.edu/~snover/tercom).
[24] The used version for METEOR is 1.5 (see http://www.cs.cmu.edu/ ~alavie/METEOR) and 1.0 for BEER (see Stanojević and Sima'an, 2014).
[25] Apart from the ones already described combinations of metrics, we can also mention VERTa (Comelles et al., 2014) and the Asiya toolkit (Giménez and Màrquez, 2010).
[26] We use the implementation made available in Python's scikit-learn library (see http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html).

overfitting, i.e. the final model should be better at generalizing on unseen test data than individual regression trees. In our case, the individual regression trees try to predict the evaluation score of the translation of the match by combining the scores of individual metrics for the source match. Highly intuitively, each tree splits the features and simply follows the branch containing the features which enable it to perform the prediction task better. Ideally, the individual trees in the forest should not be correlated, and this is partially achieved through the method of bagging, which entails taking a number of random samples from the training data, so that the individual trees are ultimately trained on different subsets of the data. In the Random Forest implementation used here, we used bootstrap sampling with replacement, which means that the same feature can be selected multiple times. A random subset of features is chosen at each split to avoid giving too much prominence to a single set of features and reduce the correlation between the trees. We trained our model on the matches retrieved for 10,000 segments by the individual metrics and used the metric scores as values of the features in training. The value the model predicts is the evaluation score of the translation of the match calculated by $METEOR_T$[27]. In addition to the individual metric scores, we added the features produced by a word2vec model[28] trained on the entire dataset without any additional text pre-processing. The model takes textual input and produces vector representations of words whose linear relationships in the vector space reportedly reflect the semantic and syntactic similarity between particular words. The vectors of individual words are averaged to obtain vector representations of segments[29]. The idea behind using the output of this model as an additional feature was that these representations might encode similarity of sentences which other metrics are unable to account for, and hence provide added value in predicting the evaluation score.

### 4.2. Automatic evaluation metrics

In order to evaluate the performance of the tested metrics on the source side, we apply automatic evaluation metrics to measure the similarity between the target side of the query

[27] The potential model parameters were roughly estimated using out-of-bag scores (i.e. by comparing the mean prediction errors on training subsets using the trees that did not include this particular subset). The optimal number of trees was found to be 700 and a maximum of 30 percent features was used at each split. The model was evaluated using 10-fold cross-validation.

[28] We used the implementation provided in Python's gensim library (see https://radimrehurek.com/gensim/models/word2vec.html). The minimal number of times a word has to occur in a corpus to be included in the dictionary was kept at 5 and the number of dimensions was limited to 100. The architecture it uses is continuous bag-of-words (CBOW).

[29] This approach of accumulating the vectors of individual words produced by the word2vec model in order to derive a vectorised sentence representation is a simple way to approximate the calculation of similarity at sentence level. Training the more advanced Sent2vec or Doc2vec models instead would probably yield more reliable results. Additional pre-processing of text is presumably also desired.

TU (the reference translation) and the target side of the retrieved fuzzy match TU. These similarity scores are then correlated to the scores produced by the original fuzzy matching metrics on the source side, to approximate the actual usefulness of the translation suggestions given to the translator by a certain metric run on the source side of the TM. Keeping in mind the self-selection bias and the relatively low scores some metrics are assigned by the faulty automatic evaluation methods, we apply four different evaluation metrics to the target side: TER, Ngram precision, METEOR and Shared partial subtrees. Out of those four, the matching configurations of $TER_T$ and $SPS_T$ are the same as on the source side. As for $NGP_T$, its parameters are set so as to in a way simulate the performance of BLEU at sentence-level. Hence, a high penalty is set for length by decreasing Z to 0. On the other hand, in order to use METEOR's synonym and paraphrase modules on the target side, we needed to develop the resources for it to use. As the Swedish version of WordNet[30] is not publicly available, we tried to develop a resource from the available version of the Swesaurus[31]. However, the relations between the items it contains are too simplistic and scarce to be integrated in METEOR's framework, and we were unfortunately unable to use this module. The paraphrase database was easier to develop and, as the research deals with EU texts, we created a parallel corpus from other available EU resources, more specifically from the English-Swedish versions of the Europarl (Koehn, 2005), EMEA (Tiedemann, 2009) and JRC-Acquis (Steinberger et al., 2006) parallel corpora. We then used this large parallel corpus to train the standard phrase-based statistical machine translation system Moses[32] (Koehn et al., 2007). As a by-product of the training process, Moses outputs phrase tables containing lexical probabilities that a particular construction could be considered as a paraphrase of another. These tables were used to create a paraphrase database using the Parex tool[33]. The database was filtered at the lexical probability threshold of 0.05 to reduce noise in the tables. Hence, $METEOR_T$ was run using the exact, stem and paraphrase module.

By choosing relatively diverse metrics for automatic evaluation, we hope to reduce the bias and obtain more realistic results. The assumption is that comparing the correlations between metrics based on subtrees, ngram spans, weighted edit distance and edit distance with shallow semantic knowledge could be interesting as each of those features is also present in some of the matching metrics, but not in others. Moreover, as we use human evaluation for this

[30] http://www2.lingfil.uu.se/ling/swn.html
[31] https://spraakbanken.gu.se/resource/swesaurus
[32] In doing this, the standard procedure for training was followed (Koehn et al., 2007), with the training, tuning and testing steps all included to increase the quality of the obtained lexical probabilities.
[33] Denkowski and Lavie (2010), Bannard and Callison-Burch (2005), see https://github.com/lixiangnlp/parex.

research, the scores assigned by the evaluation metrics themselves can also be put into perspective. Before we describe the methods for human evaluation, we first discuss the dataset and lay out the experiment procedure.

### 4.3. Data pre-processing and application of metrics

The dataset used in research is the publicly available translation memory for Acquis Communitaire provided by the Directorate-General for Translation of the European Commission (Steinberger et al., 2012). There are several practical reasons for choosing this corpus. First of all, the assumption is that this TM would provide a well-maintained and reliable dataset in terms of cleanness, reduced noise and alignment quality, but also in terms of translation quality control. As already mentioned, opting to do research with the DGT-TM also seems relevant in the light of the growing importance of TM systems and terminological resources in the complex phenomenon of translation in the context of unique multilingualism that the EU provides (Biel and Engberg, 2013; Felici, 2010). Finally, our initial idea was to conduct research for several language pairs, which made the multilingual DGT corpus a convenient choice. In the experiment, we use the English-Swedish TM, with English used as the source and Swedish as the target side, reflecting the default situation in the context of DGT translation. As a part of pre-processing, we cleaned, filtered and then parsed the data. For the English side we used Stanford parser (Klein and Manning, 2003), to which we also added lemmas. On the target side, we used the Swedish labelling pipeline efselab[34]. Both resulting parse trees were converted into the same type of xml format, nodes containing Prüfer sequences were added to them, and the monolingual parse trees were parallelised and aligned at both word and node level. The resulting parallel treebank comprised a total of nearly three million segment pairs.

This amount of data greatly exceeds our needs, but we still divide it into the training and test set to avoid overlap in the data used for different purposes. To speed up the rate of match retrieval, we index and filter the data using Approximate query coverage (Vanallemeersch and Vandegihnste 2015a), a metric which uses a suffix array (Manber and Myers, 1993) to identify segments which are likely to meet the minimal threshold set in fuzzy matching[35]. The fuzzy matching metrics are applied to these filtered results and their performance is tested for different ranges of similarity scores according to the baseline[36]. We use a subset of 10,000

[34] https://github.com/robertostling/efselab

[35] To enhance speed while still retaining as many potentially useful matches as possible, we set the filtering threshold to a value of 0.2. Other settings used in running Approximate query coverage are given in Appendix I.

[36] The main division line we base our results on in the automatic evaluation part is 0.7. The *upper range* hence denotes matches of 70 percent overlap and higher, while the *lower range* denotes matches below that score.

segment pairs as queries and compare them to the TM created from the entire test set (around 1.1 million segment pairs), applying the filter to ensure the queries would not be compared to themselves. The evaluation metrics are then applied to the target side of the retrieved matches and the two scores are correlated. As already mentioned, it is primarily the matching process in the lower similarity range that is the main focus of this research. As all the tested metrics, including the baseline, are expected to perform similarly, and reasonably well, in the highest fuzzy match range (approximately up to 80 percent overlap (Reinke, 2013)), we are interested to test the performance in the lower range where there is much more variation. The aim is to see if some of the metrics would more successfully reflect the human judgment of usefulness in this lower range, measured both in terms of ranking and score. The matches used in human evaluation were extracted from the range between 40 and 75 percent overlap. The upper bound was set slightly above the default similarity threshold, whereas the lower bound was decided by means of a manual analysis of a number of matches, as it led us to conclude that matches below this threshold would hardly be considered useful by translators in any context.

### 4.4. Human evaluation

For the human evaluation part, we created a survey using the on-line LimeSurvey platform[37], which was taken by six native Swedish speakers with training in translation[38]. We initially extracted a subset of 5,400 segment pairs from the above described dataset. As ranking fuzzy matches constitutes a demanding and laborious task, we had to further pre-process and restrict the data for human evaluation to facilitate this process. For instance, the segments chosen for ranking were filtered on length and similarity to other chosen segments, to ensure that the obtained set is relatively diverse and that the segments contained in it were long enough to be interesting and not so long as to hinder efficient comparison by human evaluators. After filtering, the dataset was further restricted to around 60 sets, consisting of the query and a maximum of six highest-scoring corresponding fuzzy matches. To facilitate the comparison, we marked the matching parts between the query and the source side of the respective matches using the baseline metric. Of course, this matching metric is coarse and faulty and the translators were warned that the mark-up was merely a reference point.

Keeping in mind the similarity range from which these matches were extracted, it would be an extremely difficult task for the evaluators to produce a full ranking of the matches, since there would hardly be many straightforward cases of particular matches being significantly

[37] An example of a survey question can be found in Appendix II.

[38] Three evaluators were professional translators working at the European Parliament and three were master students at Stockholm's Institute for Interpreting and Translation Studies.

better or worse than others. Therefore, we applied a tournament strategy to further cut down the number of matches which need to be ranked by the translators and to circumvent the need for their explicit ordering. The implementation of the tournament strategy developed for the SCATE project was based on the approach proposed by Pighin et al. (2012) and it enabled us to break down a full-ranking task into pair-wise comparisons, from which a global ranking of matches could be derived later on. In this approach, the matches in each set are organised into a tournament bracket configuration, which combines them in a way that establishes clear relationships of dominance between the matches and effectively enables the production of full rankings from relative ternary decisions: the first match is better, the second match is better or both matches are equally (un)useful as translation suggestions. For instance, if there are six fuzzy matches, the pre-terminal nodes of the tournament tree will contain three pair-wise comparisons. We constantly move to higher-level brackets by combining a random sentence from one of the lower brackets with a sentence from the other, until there is only one bracket left. In principle, this means that to derive a ranking of *n* matches, the translator needs to perform *n* (or *n+1* for an uneven number of matches) comparisons. Other filters are applied in the process to further reduce the ranking effort. The comparison results are used to build a connected graph and automatically derive a full ranking from the relative pair-wise ranks. To reduce bias, the comparisons were presented to translators in random order.

This approach was found to be faster, less laborious and more consistent in terms of achieving higher interrater agreement than approaches using explicit many-to-many comparisons directly resulting in full ranking (Green et al. 2013). The survey consists of two parts: a short section with general information questions about the translators' preferences and experience with CAT tools, and the fuzzy match evaluation part, split into two sub-sections, each comprising a total of 100 pair-wise comparisons. On the obtained data, agreement between all annotators is calculated using Fleiss' kappa, and Cohen's kappa and weighted kappa coefficients[39] are calculated between each two annotators. The scores produced by fuzzy matches on the subset are correlated to the average human evaluation derived from the survey results using Pearson correlation. The ranking of matches produced by the metrics is correlated to the normalised human ranking using Spearman's rank correlation coefficient. All results of automatic and human evaluation are presented in tables in Appendices III. and IV. and discussed in the following section.

[39] If translators prefer opposing matches, weight is 2; if one translator chose a match and the other marked both as equal, weight is 1.

## 5. Results and discussion

In this section we present and discuss the most relevant results of the experiment. First we look at automatic metrics, their correlation to the fuzzy matching metrics and mean evaluation score, and then discuss the results of human evaluation.

### 5.1. Automatic evaluation

As expected, added features rarely proved to be of significant value in the upper range of similarity overlap where the baseline is strong. Very few metrics succeeded in beating the baseline, and the margin of improvement is most of the time so slight that it is barely significant. Below we present the table with the metrics which performed better than the baseline on at least one criterion.

| | $MET_T$ corr | $MET_T$ mean | $NGP_T$ corr | $NGP_T$ mean | $SPS_T$ corr | $SPS_T$ mean | $TER_T$ corr | $TER_T$ mean |
|---|---|---|---|---|---|---|---|---|
| BASELINE | 0.4491 | 0.7516 | 0.4373 | 0.6574 | 0.4527 | 0.7554 | 0.4702 | 0.2342 |
| BEER | **0.5005** | **0.7585** | **0.5036** | **0.6663** | 0.4634 | **0.7590** | 0.4223 | 0.2341 |
| METEOR | 0.3564 | 0.7555 | 0.3666 | 0.6637 | 0.3442 | 0.7527 | 0.3123 | 0.2390 |
| $NGP_{WORD1}$ DEF | 0.473 | 0.7552 | 0.4913 | 0.6636 | 0.3701 | 0.7543 | 0.3153 | 0.2386 |
| TER | 0.4439 | 0.7521 | 0.4323 | 0.6574 | **0.4643** | 0.7562 | **0.4913** | **0.2324** |
| ALL | 0.4428 | 0.7523 | 0.4541 | 0.6589 | 0.4058 | 0.7557 | 0.4289 | 0.2344 |

Table 1: Automatic evaluation of the fuzzy matching metrics in the similarity range above or equal to 0.7. Best results are bolded. All correlations and none of the mean scores are statistically significant.

As can be seen from the table, BEER achieves better results than the baseline according to all evaluation metrics except for $TER_T$. What is also important to note is that $TER_T$ displays the strongest self-selection bias, as TER on the source side correlates best with it and achieves the best mean score. Apart from BEER, it is easy to notice that the individual metrics which achieve similar or slightly better results than the baseline in this fuzzy match range all share some of its features: METEOR and TER are essentially based on edit distance, the combination of metrics (ALL) uses both of these metrics along with the baseline as features and was optimised on $METEOR_T$ evaluation scores, and the source side NGP metric was also run on surface word forms. It is also interesting to note that METEOR, although it uses similar lexical resources as BEER, correlates poorly with all evaluation metrics, including $METEOR_T$. This leads us to conclude that lexical variability, and more generally linguistic features, does not provide added value in the highest matching range, as matching on surface forms seems to be preferred in this dataset. Apart from the similarities between the better-scoring metrics and the baseline, it is interesting to note that a degree of self-selection bias is indeed present in automatic evaluation, with only $SPS_T$ not selecting itself among the better

scoring metrics. With regard to this, the improvement over the baseline achieved by BEER indeed seems quite remarkable. Unfortunately, we can only speak of improvements on the correlation criteria with legitimacy, as none of the improvements in the mean evaluation scores are statistically significant[40]. Next we look at the range of matches whose similarity overlap is below 70 percent.

| | $MET_T$ corr | $MET_T$ mean | $NGP_T$ corr | $NGP_T$ mean | $SPS_T$ corr | $SPS_T$ mean | $TER_T$ corr | $TER_T$ mean |
|---|---|---|---|---|---|---|---|---|
| BASELINE | 0.6735 | 0.3372 | 0.5928 | 0.2502 | 0.6667 | 0.3640 | 0.6162 | 0.8687 |
| BEER | 0.6328 | 0.3499 * | 0.6031 | 0.2650 * | 0.6002 | 0.3650 | 0.4198 | 0.9962 |
| $LEV_{LEM1DEF}$ | 0.6477 | 0.3347 | 0.5559 | 0.2466 | 0.6553 | 0.3643 | 0.6202 | 0.8681 |
| METEOR | 0.6967 | 0.3534 * | 0.6723 | 0.2752 * | 0.5601 | 0.3531 | 0.3619 | 1.0049 |
| $NGP_{WORD1DEF}$ | 0.7277 | **0.3546 *** | 0.7004 | **0.2764 *** | 0.6087 | 0.3576 | 0.3587 | 1.0018 |
| $NGP_{LEM4DEF}$ | 0.6732 | 0.3358 | 0.6402 | 0.2614 * | 0.5902 | 0.3517 | 0.3935 | 0.9530 |
| $PM_{WORD1DEF}$ | 0.5577 | 0.3111 | 0.4862 | 0.2204 | 0.5116 | 0.3494 | 0.5956 | 0.7750 * |
| $PM_{LEM3DEF}$ | 0.6175 | 0.3220 | 0.5497 | 0.2379 | 0.5869 | 0.3534 | 0.5934 | 0.8259 * |
| $PM_{POS3DEF}$ | 0.4872 | 0.3008 | 0.4237 | 0.2149 | 0.4756 | 0.3410 | 0.5540 | 0.8084 * |
| $PM_{PRUF2DEF}$ | 0.4738 | 0.3018 | 0.4135 | 0.2142 | 0.4736 | 0.3436 | 0.5358 | 0.7930 * |
| SPS | 0.5625 | 0.3196 | 0.4919 | 0.2281 | 0.5804 | 0.3653 | 0.5415 | 0.8095 * |
| TER | 0.6533 | 0.3155 | 0.5841 | 0.2235 | 0.6607 | 0.3666 | **0.7075** | **0.7443 *** |
| ALL | **0.7663** | 0.3524 * | **0.7255** | 0.2687 * | **0.7089** | **0.3705 *** | 0.6566 | 0.8395 * |

Table 2: Results for fuzzy matching metrics in the similarity range below 0.7. Best results are bolded and statistically significant mean scores are marked with an asterisk. All correlations are statistically significant.

One thing that we notice straight away is the greater number of fuzzy matching metrics which performed better than the baseline according to some measure of quality in this lower range. The combination of metrics consistently correlates best with all evaluation metrics except for $TER_T$ and always outperforms the baseline. However, it is interesting to note that most of the metrics in the table only achieved improvement in the mean $TER_T$ score, which is not even necessarily reflected by their correlation with $TER_T$. Most notably, the different variants of Percent match seem to be favoured by $TER_T$ in this range. For the rest, we can notice that using linguistic information in matching arguably gets a more prominent role in this range,

[40] We measure the statistical significance of mean scores through bootstrap resampling: we take a number of query subsets and compare the mean evaluation scores of the best matches retrieved by the metrics and the baseline at the 95 percent confidence interval. We also calculate the p-value across the entire test set. The first measurement is somewhat more fine-grained, but essentially both measurements give the same results.

with metrics run on partial subtrees and sequences of lemmas, POS tags and even Prüfer representations all outperforming the baseline. Looking at both tables, we can establish that similar trends are displayed by the first three evaluation metrics across both ranges of fuzzy match score, since the same five fuzzy matching metrics are again selected as the top performing ones. $TER_T$ again displays a very strong self-selection bias, setting apart TER on the source side as by far the best performing metric. We also notice a drop in BEER's performance in the lower range across all evaluation metrics, as well as a general increase in correlations between METEOR and the first three metrics. Most importantly, we see that the baseline is still rather strong in this range, with only the combination of all metrics consistently outperforming it on all criteria.

Moreover, we notice the link between the evaluation results produced by the first two metrics, $METEOR_T$ and $NGP_T$, and the latter two, $SPS_T$ and $TER_T$. According to the first two evaluation metrics, BEER, METEOR and NGP on words perform better than the baseline. Additionally, $NGP_T$ selects the lemmatised version of NGP, which reinforces the existence of self-selection bias visible also in $METEOR_T$'s selection of METEOR. The links between $SPS_T$ and $TER_T$ are less obvious, but unlike the first two evaluation metrics, their mean scores indicate SPS and TER as the metrics outperforming the baseline. This division into two groups highlights the fact that it is unacceptable to use a single evaluation metric score as the sole estimator of quality, but unfortunately also suggests that none of the tested fuzzy matching metrics were good enough to obtain significantly improved results according to all evaluation metrics. Notably, BEER comes closest to achieving this goal, outperforming the baseline according to all evaluation metrics apart from $TER_T$ in the higher range. However, it only beats the baseline according to the first two metrics in the lower range, as the improvement in mean $SPS_T$ is not statistically significant. On a similar note, the improvements achieved by NGP on lemmas and variants of the PM should be interpreted with some caution, considering that they were rated highly only by $NGP_T$ and $TER_T$ respectively. From all four metrics, $TER_T$'s results are most difficult to interpret. It clearly favours the naïve unigram approach of the PM variants to higher-order ngrams of NGP. On the other hand, while correlating extremely well with TER and the baseline, it correlates pronouncedly poorly with METEOR as another edit-distance metric. That the self-selection bias is not always straightforwardly displayed is also visible in the fairly good $TER_T$ score of SPS. Given these results, the drop in performance of the combination of metrics when evaluated with $TER_T$ is not surprising. Overall, we note that the combination does seem to perform fairly consistently – its performance is similar to that of the baseline in the range above 70 percent

overlap and it outperforms the baseline across all evaluation metrics in the range below 70 percent.

### 5.2. Human evaluation

We now turn to the human evaluation of the metrics in the range between 75 and 40 percent overlap, to see if some of the metrics correlate better with the human judgment of usefulness than the baseline. First we need to mention that the interrater agreement between the six evaluators is relatively poor. The agreement is only 0.169 and falls into the category of slight agreement according to the kappa interpretation scale (Landis and Koch, 1977). Similarly, weighted and unweighted Cohen's kappa coefficients between each two annotators range from no agreement to fair agreement, but it is interesting to note that there is somewhat better agreement between the three professional translators. As the number of participants in the survey was not very large, we decided not to partition them further by taking into account the years of their professional experience, but the answers of the three professional translators were given slightly more weight in calculating the average scores[41]. Keeping the overall agreement in mind, let us now look at the correlations between human evaluation and the fuzzy matching metrics.

| | Spearman | Pearson |
|---|---|---|
| BASELINE | 0.3880 | 0.2087 |
| BEER | 0.4183 | 0.2407 |
| $LEV_{LEM1DEF}$ | 0.3971 | 0.2182 |
| $LEV_{LEM6IGN}$ | 0.3416 | 0.2377 |
| $LEV_{POS4DEF}$ | 0.3464 | 0.1816 |
| $LEV_{PRUF4DEF}$ | 0.3088 | 0.1464 |
| METEOR | 0.3934 | 0.1996 |
| $NGP_{WORD1DEF}$ | 0.4081 | 0.2164 |
| $NGP_{LEM3DEF}$ | 0.3768 | 0.2466 |
| $NGP_{POS4DEF}$ | 0.3620 | 0.1938 |
| $NGP_{PRUF4PRUF}$ | 0.3577 | 0.1575 |
| $PM_{WORD1DEF}$ | **0.4260** | **0.3373** |
| $PM_{LEM3DEF}$ | 0.4210 | 0.3100 |
| $PM_{POS3DEF}$ | 0.3960 | 0.2403 |
| $PM_{PRUF2DEF}$ | 0.4142 | 0.2270 |
| SPS | 0.3483 | 0.1955 |
| TER | 0.3758 | *0.1146* |
| ALL | 0.3897 | 0.2407 |

Table 3: Correlations between human evaluation and all tested metrics based on rank (Spearman's rho correlation coefficient) and score (Pearson correlation coefficient). The best results are bolded and the insignificant correlations with a p-value above 0.05 are in italics.

[41] We produced the answers of the "average evaluator" by taking the mode of the answers provided by the evaluators for each question. Giving "weight" to the answers of the professional translators hence simply meant that the mode of their answers was taken when the six answers had no mode. In the few cases when even the three of them gave completely different answers, the answer was set to 3, i.e. "both equal".

Looking at the table, it is easy to notice that the correlations overall are not very high, with the metrics generally correlating with the human judgement of usefulness better according to the criterion of rank. Although score correlations are quite low, the higher correlation in rank is in most cases reflected by an increase in the correlation in score. However, looking at examples such as NGP on lemmas, we notice that the relationship is not straightforward, as its score correlation is higher and rank correlation lower than the baseline's. Apart from the fact that the correlations are relatively low, we must not forget that the values used for correlation were derived from local pair-wise comparisons of matches and not assigned directly by the evaluators, as acquiring explicit ranks and scores would make the already demanding task even more taxing. Keeping in mind these restrictions, along with the already mentioned low interrater agreement, it is questionable with which degree of certainty we can draw conclusions from the obtained results. Even so, there are a couple of things which are interesting to note. First of all, the metric which appears to correlate best with the human judgement of usefulness according to both rank and score is the simplest one of all the tested metrics – Percent match on words. What is more, Percent match on lemmas is in the second place according to both rank and score. The performance of these rudimentary bag-of-words metrics also comes as a surprise because they did not correlate particularly well with the four automatic evaluation metrics. On the other end of the scale, TER scores correlate extremely poorly with the human judgement, and the situation is not much better with METEOR, even though both of these metrics were favoured by some of the metrics in the automatic evaluation and their complexity leads us to intuitively assume that they would perform better. Looking at the metrics with linguistic features, matching on POS-tags and Prüfer sequences using Levenshtein and NGP again scores poorly, but both of these matching items outperform the baseline when used in PM. On the other hand, all lemmatised versions of metrics have higher Pearson correlations with human judgment than the baseline. Another thing to note is the relatively poor performance of the combination of the metrics – although it outperforms the baseline, it does not perform as well as on the dataset in the automatic evaluation part. Finally, however tentative our conclusions may be, let us point out that nine of the tested configurations correlate with human judgment better than the baseline according to rank, and ten of them outperform the baseline according to score.

### 5.3. Discussion

Given the great variability of the obtained results, we take a closer look at the dataset and the matches retrieved by the metrics.

### 5.3.1. Qualitative analysis of matches

We first take the best and worst performing individual metrics from the automatically evaluated ranges and examine their output. We take 50 random sentences and extract the best matches retrieved by BEER and PM on POS-tags, as well as the best matches according to METEOR and PM on Prüfer sequences for the higher and lower range respectively. The results of the analyzed metrics across all evaluation metrics are given in the tables below for reference.

| | $MET_T$ corr | $MET_T$ mean | $NGP_T$ corr | $NGP_T$ mean | $SPS_T$ corr | $SPS_T$ mean | $TER_T$ corr | $TER_T$ mean |
|---|---|---|---|---|---|---|---|---|
| BASELINE | 0.4491 | 0.7516 | 0.4373 | 0.6574 | 0.4527 | 0.7554 | 0.4702 | 0.2342 |
| BEER | **0.5005** | **0.7585** | **0.5036** | **0.6663** | 0.4634 | **0.7590** | 0.4223 | 0.2341 |
| $PM_{POS3DEF}$ | 0.0657 | 0.7299 | 0.0380 | 0.6298 | 0.1118 | 0.7356 | 0.1960 | 0.2542 |

Table 4: Results for the matches in the range higher than or equal to 70 percent.

| | $MET_T$ corr | $MET_T$ mean | $NGP_T$ corr | $NGP_T$ mean | $SPS_T$ corr | $SPS_T$ mean | $TER_T$ corr | $TER_T$ mean |
|---|---|---|---|---|---|---|---|---|
| BASELINE | 0.6735 | 0.3372 | 0.5928 | 0.2502 | 0.6667 | 0.3640 | 0.6162 | 0.8687 |
| METEOR | 0.6967 | 0.3534* | 0.6723 | 0.2752* | 0.5601 | 0.3531 | 0.3619 | 1.0049 |
| $PM_{PRUF2DEF}$ | 0.4738 | 0.3018 | 0.4135 | 0.2142 | 0.4736 | 0.3436 | 0.5358 | 0.7930 * |

Table 5: Results for the matches in the range below 70 percent.

As expected, in the higher range most of the retrieved best scoring matches were the same for both metrics, and in this sense the extreme difference in correlations seems unjustified. The difference in mean scores is much less pronounced and according to the manual analysis can mainly be attributed to the fact that PM does not take into account the length of the matching sentence, which the evaluation metrics penalise when calculating the score on the target side:

| | |
|---|---|
| **QUERY** | Movement certificates EUR.1 or EUR-MED issued retrospectively |
| **$MATCH_{BEER}$** | Movement certificates EUR.1 issued retrospectively |
| **$MATCH_{PM}$** | Movement certificates EUR.1 or EUR-MED issued retrospectively shall be endorsed with the following phrase in English |

As the second match contains two elements more than the first one, i.e. the percentage of query elements found in the match is higher, PM chooses this match as the better one. The ability to identify overlap regardless of the segment length and word order certainly might have its advantages in some cases, but generally these matches might require substantial post-editing, especially if the words are strewn across the sentence. Moreover, even if the overlapping words are sequential in the source side match, this continuity might be disrupted on the target side in languages (and domains) with freer word order. The longer the match is, the less likely it is that such a match will be useful as a translation suggestion. On the other

hand, looking already at the above example, where the two additional elements recognised by PM are a conjunction and a term, we might also intuitively claim that not all elements should carry equal weight. Interestingly enough, in terms of the latter comment, adding features such as IDF weights to PM does not seem to necessarily yield better results (Vanallemeersch and Vandeghinste, 2015a), whereas in terms of the former we have to keep in mind that all versions of PM outperformed the baseline according to human evaluation.

In the lower range, we look at the matches retrieved by METEOR as it is among the best scoring metrics according to $METEOR_T$ and $NGP_T$, but scores less well with $SPS_T$ and extremely poorly according to $TER_T$. We also look at PM with Prüfer sequences, as its performance is below the baseline according to all criteria, except for the mean $TER_T$ score, which is significantly better than the baseline's. Even in this small extracted subset, there are matches whose scores are so low that both best sentences retrieved by the metrics seem random to us and are useless as translation suggestions, so setting a bottom threshold to limit the overlap range might have yielded more realistic results in automatic evaluation. The suggestions made by the PM metric are again generally longer and this could have an unfavourable effect on its evaluation by the automatic metrics. On the other hand, there is again a significant number of overlapping sentences retrieved by both metrics, and sometimes the PM metric actually retrieves a slightly better suggestion according to our subjective judgment:

| | |
|---|---|
| **QUERY** | Commission Decision 2005/392/EC of 17 May 2005 amending Decision 2004/233/EC as regards the list of laboratories authorised to check the effectiveness of vaccination against rabies in certain domestic carnivores is to be incorporated into the Agreement. |
| **$MATCH_{MET}$** | amending Decision 2004/233/EC as regards the list of laboratories authorised to check the effectiveness of vaccination against rabies in certain domestic carnivores |
| **$MATCH_{PM}$** | Commission Decision 2005/656/EC of 14 September 2005 amending Decision 2004/233/EC in terms of the laboratories authorised to check the effectiveness of vaccination against rabies in certain domestic carnivores is to be included in the Agreement |

However, low scoring matches such as these are even more pronouncedly subject to the external factor of translator's preference: whether he or she will rather take over the entire METEOR match and fill in the missing parts, take the PM match and edit the differences, combine the useful parts of both or use neither because the time he or she needs to translate from scratch is shorter than time needed for post-editing a poor match is influenced by a number of factors. These matters are briefly addressed in the next subsection.

The second approach to doing manual analysis focused on the matches ranked by the translators in the survey. We extract the matches whose baseline score differs radically from the evaluation score, i.e. the segments where the source side match got a much higher score

than the target side match and vice versa. We begin with two very simple examples illustrating the latter case.

| | | |
|---|---|---|
| **$Q_1$** | SS | Insurance corp. and pension funds |
| | TS | Försäkringsföretag och pensionsinstitut |
| **$M1_1$** | SS | Insurance corporations and pension funds and pension funds |
| | TS | Försäkringsföretag och pensionsinstitut |
| **$M2_1$** | SS | Insur. corporations and pension funds |
| | TS | Försäkringsföretag och pensionsinstitut |
| **$Q_2$** | SS | Transport, storage and communications |
| | TS | Transport, maganisering och kommunikation |
| **$M_2$** | SS | Transports, storage and communication |
| | TS | Transport, maganisering och kommunikation |

According to the baseline, both of the matches in the first example get a score of 0.5 and the match in the second example gets a score of 0.6. While both scores are below the standard 0.7 threshold, we notice that the matches on the target side are in fact exact matches. However, in practice, the translators would not even be offered these perfect translation suggestions because of the mistakes and minor differences on the source side greatly affecting the calculated fuzzy match score. In this small dataset, the cases where the source side got a much higher score than the target side are more frequent more diverse:

| | | |
|---|---|---|
| **$Q_1$** | SS | The modalities of the certificate shall be decided by the Steering Committee. |
| | TS | Villkoren för intyget skall fastställas av styrkommittéen. |
| **$M1_1$** | SS | The financing of the PE shall be decided by the JIC. |
| | TS | Finansieringen av periodiska utvärderingar ska beslutas av den gemensamma kommittéen för genomförandet av avtalet. |
| **$M2_1$** | SS | The convocation of such conference shall be decided by the Council. |
| | TS | Rådet skall besluta om sammankallandet av en sådan konferens. |
| **$Q_2$** | SS | The proceedings shall be circulated after each meeting. |
| | TS | Protokoll skall skickas ut efter varje möte. |
| **$M_2$** | SS | The decisions and recommendations shall be circulated to the Parties. |
| | TS | Besluten och rekommendationerna ska spridas till parterna. |
| **$Q_3$** | SS | The health certificate must be presented to the competent veterinary authorities at the request of the latter; |
| | TS | Detta hälsointyg skall på begäran kunna visas upp för de behöriga veterinärsmyndigheterna. |
| **$M_3$** | SS | The signed certificate must be forwarded to the competent authority at the place of physical check. |
| | TS | Det undertecknade intyget måste överlämnas till behörig myndighet på den plats där den fysiska kontrollen genomförs. |

In the first two examples, we notice that if we were only to look at the source side query and the respective target side matches, it would be difficult to see the reason why these would be offered as potential translation suggestions at all. On the source sides of these matches, we see continuous strings of overlap, but these are either not particularly valuable or not even present on the target side. This highlights the fact that Levenshtein on words is not a good estimator of quality for lower score ranges and that attempting to improve the system by lowering this

metric's threshold to increase the recall probably would not yield satisfactory results. The last segment is a good illustration of how linguistic differences affect the segment's usefulness: the already sparse overlapping elements on the source side are rendered useless on the target side by being agglutinated into compounds. That is, the overlapping *certificate* is translated as *hälsointyg* and *intyget*, whereas *authorities* become *veterinärsmyndigheterna* and *myndighet* respectively. The issue of how to deal with these phenomena can be considered from the opposite point of view, i.e. how do we take into account the fact that *myndighet* and *myndigheterna* only differ in number and definite form, and that a *hälsointyg* is still in fact a type of *intyg*. Here, however, we focus on the flectional phenomena and the agglutination process resulting in higher post-editing effort and consequently reducing the usability of the match.

### 5.3.2. Translators' notion of usefulness

As the explicit investigation of translators' preferences and expectations from TM systems is not the focus of this research, we will only briefly discuss this matter in relation to the General information section included in the survey. More precisely, we will look at the mean ratings of the features the translators (in theory) consider important in fuzzy matches. The translators were asked to rate the importance of five features characterizing the offered matches on a scale from 1 (not important) to 5 (very important). These features highlight the aspects that the tested metrics are purportedly better at capturing than the baseline. According to the average rating, the most important characteristic (4.33) is for the match to contain specific terminology or named entities. This is a surface-level feature that the baseline can capture, but we might expect its lemmatised variant to be more successful at it, and maybe even NGP or PM on words or lemmas, since the former should be better at capturing phrase-like ngram structures and the latter should do well with identifying smaller units inside sentences, as it does not take into account word order in score calculation. The next feature referred to editing effort in terms of preferring longer continuous overlapping spans and shorter sentences in cases where matching phrases are discontinuous. It got an average rating of 3.67 and implies preferring NGP to PM metrics. All edit-distance-based metrics should also perform well on this task.

Next we have two features aimed at phenomena which are above the lexical level: matches which share the same meaning or the same syntactic patterns with the query, but differ in the actual wording. As the lack of an adequate way to deal with semantics is frequently pointed out as the main disadvantage of TM and MT systems in general, it is somewhat surprising that it only got a rating of 3.0. On the other hand, the importance of overlap in syntactic patterns is

expectedly low (2.33) if we consider the nature of the translation task. The computationally more sophisticated among the tested metrics (e.g. BEER, METEOR, SPS and maybe TER[42]) are expected to perform better than the baseline in capturing these two aspects, but the idea was that even the simpler metrics run on POS-tags and Prüfer sequences might produce interesting results. Finally, the last feature does not give advantage to any particular metric over the baseline, but was rather used as an indication of the translator's habit in terms of balancing precision and recall. The importance of the percentage of overlap got a rating of 4.17, and if we look at the values the translators gave for the threshold they usually use in translation (mostly around 65[43]), this effectively favours metrics whose output has higher precision. Looking at the translators' answers and the correlation table with the fuzzy matching metrics, it does stand to reason that the string-based, surface-level metrics still get the upper hand, regardless of our initial intuitive assumption being different. However, the good performance of the high-recall PM metrics is still somewhat surprising.

Finally, several notes and comments are in order regarding the experimental setup itself, which qualitative analysis of the data and the survey results have brought into to focus.

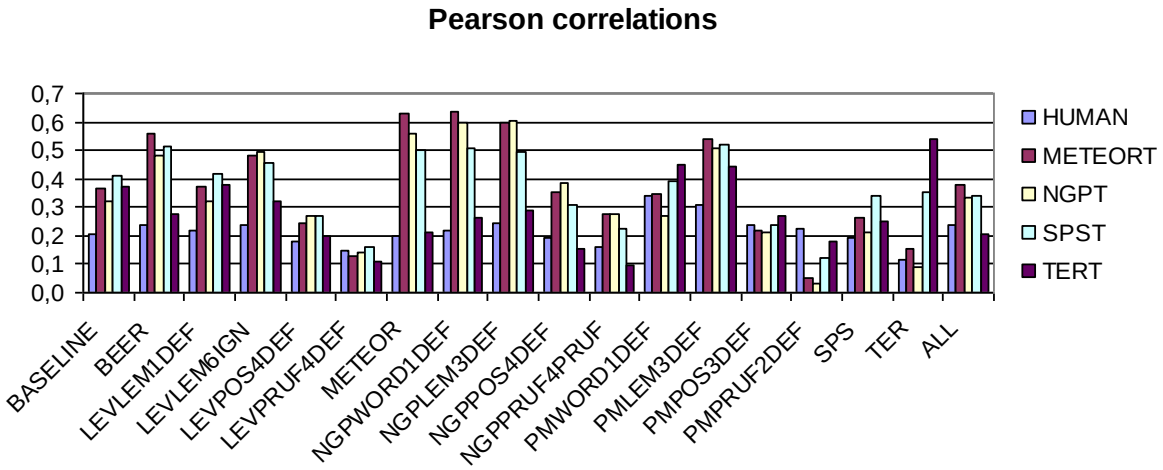


Figure 1: Pearson correlations between the fuzzy matching metrics and human evaluation and automatic evaluation metrics.

The Pearson correlations presented in the figure above were calculated on the small subset evaluated by the translators in the survey[44]. The first thing we notice is that, apart from METEOR and some of the PM variants, it is mostly the same metrics which outperform the

[42] The implementation used in this research does not use any additional lexical resources.
[43] It is interesting to mention that the translator who gave 80 as the preferred threshold value gave us feedback on the survey, saying that he apologises if his answers were not of much use, as he really only uses nearly perfect matches and the rest translates quicker from scratch. This again highlights the fact that the results of research such as this one are highly relativised when taking into account the actual preferences of individual end users.
[44] The full correlations table can be found in Appendix IV.

baseline according to human evaluation and the first three automatic metrics. However, important to note here is the fact that the correlations of the first three automatic metrics are generally much higher than those displayed by $TER_T$ and human evaluation. If we hence conclude that $TER_T$ scores might indeed be a better approximation of the human notion of usefulness, the question arises if the combination of metrics would correlate better with human judgment had we used $TER_T$ in training the regression model. This need not be the case, as $TER_T$ consistently strongly correlates best with itself, which would probably result in the model giving a lot of prominence to this feature. To get more realistic results, we would then probably have to exclude TER as a feature, something we did not find necessary in the current model setup, with the correlation between METEOR and $METEOR_T$ being much less pronounced. Generally speaking, considering using a combination of the target side metrics maybe would have been better for model training, as this would hopefully also make the model less tuned to this particular dataset, than retrospectively opting for a different single evaluation metric after manually analysing the results and output. This brings us to another point which came out of our qualitative analysis, and that is the idea of using the target side of the dataset to somehow inform the matching process on the source side. This would make sure that the similarity measured on the source side is actually retained on the target side, since it is this similarity the translator is ultimately interested in. However, this brings out a number of problematic underlying assumptions when doing computational research. Namely, when doing research with extensive amounts of data, one has to assume that the data, after filtering and pre-processing, is perfect. This is of course hardly the case and problems are likely to arise at every level, from crude mistakes such as misalignment and faulty segmentation, to more sophisticated mistakes produced by parsers and the matching algorithms themselves. Naturally, the potential risk of something going wrong only increases the more complex we make the matching process and the more languages we include.

## 6. Conclusion

In this thesis we examined the idea that the inclusion of linguistic features in fuzzy matching might improve the functioning of the existing TM systems. The intuitive assumption is that establishing and measuring similarity between two segments of text based only on the exact word forms and word order is insufficient to capture many levels of similarity as perceived by humans. Considering that most commercial TM systems still seem to use some variants of the simple surface-level edit-distance matching metric, we tested whether metrics run on different elements than word forms or using additional linguistic

resources could retrieve better translation suggestions according to both automatic and human evaluation. As all metrics regardless of their complexity and specific features perform similarly in the highest fuzzy match range (approximately up to 80 percent overlap), the research was focused on the improvement of the performance of the metrics in the matching range below the 70 percent threshold, i.e. on the matches translators would not even be offered as suggestions in a default translation situation.

The fuzzy matching framework developed within the SCATE project enabled us to include a diversity of linguistic features contained in the created parse trees in the matching process: lemmas, part-of-speech tags, subtree structures and tree structures "flattened" into Prüfer sequences. We also experimented with additional synonym and paraphrase resources. To diversify the metrics even more, we experimented with a number of weighting schemes and ngram orders, before setting the final matching configurations. Looking at the results, BEER is arguably the one metric which stands out, as its correlations and mean evaluation scores generally exceed the baseline's. The combination of metrics created using the Random Forest Regressor also performs well according to the automatic metrics, achieving by far the best results in the lower fuzzy match range. However, its correlation with the human judgment is much lower, although it still exceeds the baseline. Overall, more metrics outperform the baseline according to human evaluation, with the different variants of the simple Percent match metric correlating strikingly well with human scores and ranks. The poor correlations of METEOR and TER are also surprising, but we should keep in mind the limitations pertaining to this part of the research: the small number of evaluators, the limited amount of work that the humans can be expected to perform in comparison to the size of the entire dataset and, most importantly, the low agreement between the evaluators. The obtained results show that the notion of usefulness of matches in this range is highly dependant on individual preferences. It would therefore be somewhat strained to claim that using POS-tags or Prüfer sequences generally improves the quality of fuzzy matching, especially as Levenshtein and NGP run on these match items correlate with human judgment worse than the baseline.

According to automatic evaluation, it is primarily the more sophisticated among the tested metrics that come close to (or outperform) the baseline in the matching range above 70 percent overlap. More metrics are successful at beating the baseline in the lower range, with the PM variants achieving significant improvement on the mean $TER_T$ score. This would lead us to conclude that using linguistic features really does have added value in cases where the baseline performs poorly. However, the bias towards favouring the similarly functioning source side metric is to various degrees visible in all evaluation metrics apart from $SPS_T$, so

we would once again like to point out BEER as the only individual metric to more consistently achieve improved results over the baseline. Incidentally, it does in a way prove the worth of linguistic features in matching, as alongside using character-based ngrams, BEER also uses syntactic features in the form of node permutations and information on the distinction between content and function words, as well as a number of lexical resources to identify similarity in meaning.

This brings us back to a number of problematic issues mentioned at the beginning of the thesis. First of all, as mentioned in the literature overview, BEER was primarily developed as an MT evaluation metric and not as a fuzzy matching metric. This is important because speed can be sacrificed to a much greater extent in MT evaluation in order to ultimately obtain better results. BEER is computationally extremely heavy and, in practice, if the match is not offered to a translator almost instantaneously, it will hardly be very useful, no matter how good a translation suggestion it may constitute, and most definitely will not result in speeding up the translation process. This problem could be partly resolved by the pre-processing, indexation and matching being done before the translation begins, but this would put a considerable amount of strain on the preparatory step, and the on-line updating of translation memories and other CAT tool features (such as termbases and MT systems) would have to somehow be dealt with. This drastic drop in speed when using more sophisticated similarity algorithms might be one of the reasons why CAT tools still persist with using edit distance. Another reason might lie in the fact that using linguistic information in matching requires language-specific resources and tools. On that note, BEER might work well for English, but the lexical resources it draws on have been developed for very few languages. Moreover, not all languages can be easily integrated into a framework based on parse trees, even if there are parsers available for them. Even though English and Swedish belong to the same language family and share a similar linguistic tradition, and even though both languages are well-covered in terms of the developed resources and tools, we still encountered numerous issues in trying to incorporate them into a single framework. Not only is the format of the output of particular parsers specific, the very logic on which the parsers are built may be very different, which might give rise to a number of obstacles when trying to apply the same approach to two different languages.

At this point we must also mention that the entire setup should be further tested on a different language pair, as it would be especially interesting to see if using linguistic features might have a more significant impact when doing matching on morphologically rich languages. Unfortunately, even though we intended to examine this matter as well by

including Croatian in the research, a number of setbacks and issues regarding the integration of Croatian into the framework made it impossible to carry out this plan. This initial idea of having three languages also resulted in the fact that we only did research with a single dataset, as the DGT-TM is available for all three languages and the uniformity of the domain of the corpus the tests are run on would give some consistency to the obtained results. It is therefore still very much a question if the individual metrics and their combination would be equally (un-)successful if applied, for instance to corpora whose language is much less restrained than the legal language of the DGT dataset[45]. More particularly, we might wonder whether a metric such as METEOR would perform better on types of text more prone to lexical diversity, or would a tree-based metric such as SPS perform worse on texts with much freer syntax structure. All these issues remain as points for further investigation.

Regarding our research, we can conclude that Levenshtein on word sequences provides a fairly strong baseline for this dataset. Although some of the tested metrics are built on very interesting, and generally highly intuitive ideas, very few of them succeeded in beating the baseline in the highest matching range, which at this point still makes their implementation in the CAT tools "uncalled for", as they would potentially make the matching process considerably slower and more complex because of the language-specific features, while not making it significantly better. In the lower range, more metrics outperformed the baseline according to a number of measurements of quality, but the question still remains whether the translators would truly consider them useful as translation suggestions in an actual translation situation and how the slightly, or considerably, lower-scoring metrics could best be integrated into the CAT tool environments to utilise their advantages and reduce the post-editing effort. Despite some promising results, the question of fuzzy matching and automatic evaluation metrics still very much remains an unsolved problem, but we can hope that the matter will soon start getting more attention in the commercial sphere, instead of just be a matter of interest to the research community.

[45] The findings of Gupta et al. (2014b), who used an SVM model to calculate and combine a wide variety of linguistic and non-linguistic similarity features, back this claim, as they acheived significant improvement over the baseline on Europarl, but their model did not beat the baseline on the DGT dataset. Same goes for Gupta et al. (2016), who report the added value of the paraphrase resources they enhanced the matching metric with in their experiment was much lower for the DGT dataset.

**References**

Baldwin, Timothy (2010). The Hare and the Tortoise: Speed and Accuracy in Translation Retrieval. *Machine Translation,* 23(4): 195–240.

Bannard, Colin and Chris Callison-Burch (2005). Paraphrasing with Bilingual Parallel Corpora. *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics.*

Banarjee, Satanjeev and Alon Lavie (2005). METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization,* 65–72.

Biel, Łucja and Jan Engberg (2013). Research models and methods in legal translation. *Linguistica Antverpiensia, New Series - Themes in Translation Studies 12 (2013).* Web. 10 Nov 2017.

Blésius, Corinne (2003). Copyright and the Translator: Who Owns your Translations? *ITI Bulletin. The Journal of the Institute of Translation and Interpreting, November–December 2003,* 9–12.

Bloodgood, Michael and Benjamin Strauss (2014). Translation memory retrieval methods. *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, 202–210.

Bär, Daniel, Chris Biemann, Iryna Gurevych and Torsten Zesch (2012). Ukp: Computing semantic textual similarity by combining multiple content similarity measures. *First Joint Conference on 788 Lexical and Computational Semantics, Association for Computational Linguistics,* 435–440.

Christensen, Tina Paulsen and Anne Gram Schjoldager (2010). Translation-memory (TM) research: What do we know and how do we know it? *Hermes – The Journal of Language and Communication, 44,* 1-13.

Comelles, Elisabet, Jordi Atserias, Victoria Arranz, Irene Castellón and Jordi Sesé (2014). VERTa: Facing a Multilingual Experience of a Linguistically based MT Evaluation. *Proceedings of the 9th International Conference on Language Resources and Evaluation,* Reykjavik, Iceland, 2701–2707.

Denkowski, Michael and Alon Lavie (2010). METEOR-NEXT and the METEOR Paraphrase Tables: Improved Evaluation Support For Five Target Languages. *Proceedings of the ACL 2010 Joint Workshop on Statistical Machine Translation and Metrics MATR.*

Denkowski, Michael and Alon Lavie (2014). Meteor Universal: Language Specific Translation Evaluation for Any Target Language. *Proceedings of the 9th Workshop on Statistical Machine Translation,* Baltimore, Maryland, USA, 376–380.

Drugan, Jo and Bogdan Babych (2010). Shared Resources, Shared Values? Ethical Implications of Sharing Translation Resources. In Ventsislav Zhechev (ed.*): Proceedings of the Second Joint EM+/CNGL Workshop "Bringing MT to the User: Research on Integrating MT in the Translation Industry" (JEC '10),* Denver, CO, 3–9.

Federico, Marcello, Alessandro Cattelan and Marco Trombetti (2012). Measuring User Productivity in Machine Translation Enhanced Computer Assisted Translation. *Proceedings of AMTA 2012.*

Felici, Annarita (2010). Translating EU law: Legal issues and multiple dynamics. *Perspectives: Studies in Translatology, 18(2),* 95–108.

Gautam, Shubham and Pushpak Bhattacharyya (2014). LAYERED: Metric for Machine Translation Evaluation. *Proceedings of the 9th Workshop on Statistical Machine Translation,* Baltimore, Maryland, USA. 387-393.

Giménez, Jesús and Lluís Màrquez (2007). Linguistic Features for Automatic Evaluation of Heterogenous MT Systems. *ACL Workshop on Statistical Machine Translation.*

Giménez, Jesús and Lluís Màrquez (2010). Asiya: An Open Toolkit for Automatic Machine Translation (Meta-)Evaluation. *The Prague Bulletin of Mathematical Linguistics, 94:* 77–86.

Green, Spence, Jeffrey Heer and Christopher D. Manning (2013). The Efficacy of Human Post-Editing for Language Translation. *ACM Human Factors in Computing Systems (CHI).*

Gupta, Rohit, Hanna Bechara, Ismail El Maarouf, and Constantin Orăsan (2014a). UoW: NLP techniques developed at the University of Wolverhampton for Semantic Similarity and Textual Entailment. *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval-2014).*

Gupta, Rohit, Hanna Bechara and Constantin Orăsan (2014b). Intelligent Translation Memory Matching and Retrieval Metric Exploiting Linguistic Technology. *Proceedings of Translating and the Computer 36,* London, UK. 86–89.

Gupta, Rohit, Constantin Orăsan, Qun Liu and Ruslan Mitkov (2016). A Dynamic Programming Approach to Improving Translation Memory Matching and Retrieval Using Paraphrases. In Sojka P., Horák A., Kopeček I., Pala K. (eds). *Text, Speech, and Dialogue. TSD 2016. Lecture Notes in Computer Science, vol 9924,* 259–269. Springer, Cham.

He, Yifan, Yanjun Ma, Andy Way and Josef Van Genabith (2010). Integrating n-best smt outputs into a tm system. *Proceedings of the 23rd International Conference on Computational Linguistics: Posters,* 374–382.

Hodász, Gábor and Gábor Pohl (2005). MetaMorpho TM: a linguistically enriched translation memory. *Proceedings of the workshop: Modern Approaches in Translation Technologies 2005*, Borovets, Bulgaria, 25–30.

House, Juliane (2000). Quality of translation. In M. Baker (ed.). *Routledge encyclopedia of Translation Studies. 2.reimpr. (1.ed. 1998).* London: Routledge. 197- 200.

Jiang, Tao, Lushen Wang, and Kaizhong Zhang (1995). Alignment of Trees – An Alternative to Tree Edit. *Theoretical Computer Science,* 143(1): 137-148.

Jurafsky, Daniel and James H. Martin (2009). *Speech and Language Processing (2nd ed).* New Jersey: Pearson education.

Klein, Dan and Christopher Manning (2003). Fast Exact Inference with a Factored Model for Natural Language Parsing. *Advances in Neural Information Processing Systems 15 (NIPS), MIT Press.* 3–10.

Klein, Philip (1998). Computing the Edit Distance between Unrooted Ordered Trees. *Proceedings of the 6th Annual European Symposium on Algorithms,* Venice, Italy. 91–102.

Koby, Geoffrey S., Paul Fields, Daryl Hague, Arle Lommel and Alan Melby (2014). Defining Translation Quality. *Traducció i qualitat 14. Revista Tradumàtica: tecnologies de la traducció,* 413-420.

Koehn, Philipp (2005). Europarl: A Parallel Corpus for Statistical Machine Translation. *Proceedings of the 10th Machine Translation Summit,* Phuket, Thailand, 79–86.

Koehn, Philipp, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, ... and Chris Dyer (2007). Moses: Open source toolkit for statistical machine translation. *Proceedings of the 45th annual meeting of the Association of Computational Linguistics,* 177-180.

Koehn, Philipp and Jean Senellart (2010). Convergence of translation memory and statistical machine translation. *Proceedings of AMTA Workshop on MT Research and the Translation Industry,* 21–31.

Lagoudaki, Pelagia Maria (2009). *Expanding the Possibilities of Translation Memory Systems.* Doctoral dissertation. University of London.

Landis, J. Richard and Gary Koch (1977). The measurement of observer agreement for categorical data. *Biometrics 33:*159–74.

Levenshtein, Vladimir I. (1966). Binary Codes Capable of Correcting Deletions, Insertions, and Reversals. *Soviet Physics Doklady*, 10(8):707–710.

Li, Guoliang, Xuhui Liu, Jianhua Feng, and Lizhu Zhou (2008). Efficient Similarity Search for Tree-Structured Data. *Proceedings of the 20th International Conference on Scientific and Statistical Database Management*, Hong Kong, China. 131–149.

Liu, Ding and Daniel Gildea (2005). Syntactic Features for Evaluation of Machine Translation. *Proceedings of ACL 2005 Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, Ann Arbor, Michigan, USA. 25–32.

Lo, Chi-kiu and Dekai Wu (2011). MEANT: An Inexpensive, High-accuracy, Semi-automatic Metric for Evaluating Translation Utility via Semantic Frames. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language echnologies – Volume 1*, Portland, Oregon, USA, 220–229.

Ma, Yanjun, Yifan He, Andy Way, and Josef van Genabith (2011). Consistent Translation using Discriminative Learning: a Translation Memory-inspired Approach. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies – Volume 1*, Portland, Oregon, 1239–1248.

Manber, Udi and Gene Myers (1993). Suffix Arrays: A New Method for On-line String Searches. *SIAM Journal on Computing, 22:*935–948.

Moorkens, Joss and O'Brien, Sharon. 2017. "Assessing User Interface Needs of Post-Editors of Machine Translation". In Kenny, Dorothy (ed.), *Human Issues in Translation Technology: The IATIS Yearbook.* Abingdon: Routledge. 127-148.

Owczarzak, Karolina, Josef van Genabith and Andy Way (2007). Evaluating machine translation with LFG dependencies. *Machine Translation, 21*(2): 95–119.

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu (2002). BLEU: a Method for Automatic Evaluation of Machine Translation. *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, Philadelphia, Pennsylvania, USA. 311–318.

Parra Escartín, Carla (2015). Creation of new TM segments: Fulfilling translators' wishes. *Proceedings of the Workshop on Natural Language Processing for Translation Memories (NLP4TM)*, Hissar, Bulgaria, 1–8.

Pekar, Viktor and Ruslan Mitkov (2007). New Generation Translation Memory: Content-Sensitive Matching. *Proceedings of the 40th Anniversary Congress of the Swiss Association of Translators, Terminologists and Interpreters*.

Pighin, Daniele, Lluís Formiga and Lluís Màrquez (2012). A Graph-based Strategy to Streamline Translation Quality Assessments. *Proceedings of AMTA 2012.*

Prüfer, Heinz. 1918. Neuer Beweis eines Satzes über Permutationen. *Archiv der Mathematik und Physik,* 27: 742–744.

Pym, Anthony (2003). Translational ethics and electronic technologies. *Proflssionalização do Tradutor, Lisboa: Fundação para a Ciência ea Tecnologia / Uniao Latina, 2004,* 121-12.

Pym, Anthony (2006). Translation technology as rupture in the philosophy of dialogue. *Proceedings of the sixth Portsmouth Translation Conference on Translation Teclwologies and Cultures, 2006*.

Reinke, Uwe (2013). State of the art in translation memory technology. *Translation: Computation, Corpora, Cognition, 3*(1), 27-48.

Seal, Thomas (1992). ALPNET and TSS: The commercial realities of using a computeraided translation system. *Translating and the Computer 13. Proceedings from the Aslib Conference 1991,* 120-125.

Sikes, Richard (2007). Fuzzy Matching in Theory and Practice. *Multilingual,* 18(6): 39–43.

Simard, Michel and Atsushi Fujita (2012). A Poor Man's Translation Memory Using Machine Translation Evaluation Metrics. *Proceedings of the 10th Conference of the Association for Machine Translation in the Americas,* San Diego, California, USA.

Snover, Matthew, Bonnie Dorr, Richard Schwartz, Linnea Micciula and John Makhoul (2006). A study of translation edit rate with targeted human annotation. *Proceedings of the Association for Machine Translation in the Americas* (Vol. 200, No. 6).

Stanojević, Miloš and Khalil Sima'an (2014). BEER: BEtter Evaluation as Ranking. *Proceedings of the 9th Workshop on Statistical Machine Translation,* Baltimore, Maryland, USA. 414–419.

Steinberger Ralf, Bruno Pouliquen, Anna Widiger, Camelia Ignat, Tomaž Erjavec, Dan Tufis 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 (LREC'2006).* Genoa, Italy, 24-26 May 2006.

Steinberger Ralf, Andreas Eisele, Szymon Klocek, Spyridon Pilos and Patrick Schlüter (2012). DGT-TM: A freely Available Translation Memory in 22 Languages. *Proceedings of the 8th international conference on Language Resources and Evaluation (LREC'2012),* Istanbul, 21-27 May 2012.

Tiedemann, Jörg (2009). News from OPUS - A collection of multilingual parallel corpora with tools and interfaces. In N. Nicolov, K. Bontcheva, G. Angelova & R. Mitkov (eds.). *Recent Advances in Natural Language Processing*, Amsterdam: John Benjamins, 237-248.

Timonera, Katerina and Ruslan Mitkov (2015). Improving Translation Memory Matching through Clause Splitting. *Proceedings of the Workshop on Natural Language Processing for Translation Memories (NLP4TM)*, Hissar, Bulgaria, 17–23.

Vanallemeersch, Tom and Vincent Vandeghinste (2015a). Assessing linguistically aware fuzzy matching in translation memories. *Proceedings of EAMT 2015*, 153-160.

Vanallemeersch, Tom and Vincent Vandeghinste (2015b). Semantics-based pretranslation for SMT using fuzzy matches. *Proceedings of SSST, NAACL-HLT 2015 Workshop on Syntax, Semantics and Structure in Statistical Translation*, 61-64.

White, John S., Theresa O'Connell and Lynn Carlson (1993). Evaluation of Machine Translation. In Human Language Technology. *Proceedings of a Workshop (ARPA)*, 206–210.

Wolff, Friedel, Laurette Pretorius, Loïc Dugast and Paul Buitelaar (2016). Self-selection bias of similarity metrics in translation memory evaluation. *Machine Translation 30:* 129.

Zhechev, Ventsislav and Josef van Genabith (2010). Maximising TM Performance through Sub-Tree Alignment and SMT. *Proceedings of the 9th conference of the Association for Machine Translation in the Americas*, Denver, Colorado, USA.

## Appendices

### Appendix I.

### Matching algorithms and configurations

Formulae for the calculation of similarity between the query (Q) and the source side of the match (S), or the query and its reference translation (R) in case of MT evaluation metrics. The formulae which are not given here can be found in the references given in thesis.

**Levenshtein distance:**

$LEV(Q,S_i) = 1 – (\Delta_{LEV}(Q,S_i) / \max(|Q|,|S_i|))$

**Percent match:**

$PM(Q,S_i) = |Q_{unigrams}| \cap |S_{i,unigrams}| / |Q_{unigrams}|$

**Ngram precision:**

$NGP = \Sigma_{n=1}^{N} (|Q_{n\text{-}grams} \cap S_{i,n\text{-}grams}|) / ( Z * |Q_{n\text{-}grams}| \mid (1\text{-}Z) * |S_{i,n\text{-}grams}|)/N$

**Normalised TER:**

$TER(Q,R) = 1 – (\log(1+\Delta_{TER}(Q,R) / |R|))/3)$

| METRIC | MATCH ITEM | WEIGHTING SCHEME | SPECIAL PARAMETERS |
|---|---|---|---|
| Approximate query coverage | word part sequences, 1 | default | threshold: 0.2<br>nbest: 50 |
| Baseline (Levenshtein) | words, 1 | default | / |
| BEER | words, 1 | default | all modules for EN |
| Levenshtein | lemmas, 1 | default | / |
| Levenshtein | lemmas, 6 | ignore case | / |
| Levenshtein | POS-tags, 4 | default | / |
| Levenshtein | Prüfer sequences, 4 | default | / |
| METEOR | words, 1 | default | all modules for EN |
| Ngram precision | words, 1 | default | N = 4, Z =0.3 |
| Ngram precision | lemmas, 4 | default | N = 4, Z =0.3 |
| Ngram precision | POS-tags, 4 | default | N = 4, Z =0.3 |
| Ngram precision | Prufer sequences, 4 | Prüfer weights | N = 4, Z =0.3 |
| Percent match | words, 1 | default | / |
| Percent match | lemmas, 3 | default | / |
| Percent match | POS-tags, 3 | default | / |
| Percent match | Prüfer sequences, 2 | default | / |
| Shared partial subtrees | parse | default | / |
| TER | words, 1 | default | / |
| $METEOR_T$ | words, 1 | default | exact, stem, paraphrase modules for SE |
| Ngram $precision_T$ | words, 2:1 | default | N=4, Z =0 |
| Shared partial $subtrees_T$ | parse | default | / |
| $TER_T$ | words, 1 | default | / |

Table 1: All filtering, matching and evaluation configurations.

## Appendix II.

An example from the main group of survey questions.

***The agenda shall be adopted by the Trade Committee at the beginning of each meeting.**

**This question is mandatory.**

- ○ < The > budget < shall be adopted by > < the > Commission.

  Budgeten ska antas av kommissionen.
- ◉ < The > final < agenda shall be adopted > < at the beginning of each meeting. >

  Den slutliga dagordningen skall antas i början av varje sammanträde.
- ○ Both equal

## Appendix III.

### Automatic evaluation of the data

| | $MET_T$ corr | $MET_T$ mean | $NGP_T$ corr | $NGP_T$ mean | $SPS_T$ corr | $SPS_T$ mean | $TER_T$ corr | $TER_T$ mean |
|---|---|---|---|---|---|---|---|---|
| BASELINE | 0.4491 | 0.7516 | 0.4373 | 0.6574 | 0.4527 | 0.7554 | 0.4702 | 0.2342 |
| BEER | **0.5005** | **0.7585** | **0.5036** | **0.6663** | 0.4634 | **0.7590** | 0.4223 | 0.2341 |
| $LEV_{LEM1DEF}$ | 0.2948 | 0.7442 | 0.2569 | 0.6465 | 0.3475 | 0.7514 | 0.4006 | 0.2392 |
| $LEV_{LEM6IGN}$ | 0.2801 | 0.7447 | 0.2541 | 0.6481 | 0.3076 | 0.7507 | 0.3349 | 0.2411 |
| $LEV_{POS4DEF}$ | 0.0737 | 0.7320 | 0.0479 | 0.6325 | 0.1242 | 0.7372 | 0.2103 | 0.2518 |
| $LEV_{PRUF4DEF}$ | 0.1934 | 0.7484 | 0.1837 | 0.6524 | 0.2477 | 0.7512 | 0.2581 | 0.2380 |
| METEOR | 0.3564 | 0.7555 | 0.3666 | 0.6637 | 0.3442 | 0.7527 | 0.3123 | 0.2390 |
| $NGP_{WORD1DEF}$ | 0.473 | 0.7552 | 0.4913 | 0.6636 | 0.3701 | 0.7543 | 0.3153 | 0.2386 |
| $NGP_{LEM4DEF}$ | 0.3131 | 0.7438 | 0.2909 | 0.6475 | 0.3106 | 0.7480 | 0.3132 | 0.2456 |
| $NGP_{POS4DEF}$ | 0.0897 | 0.7320 | 0.0691 | 0.6325 | 0.1275 | 0.7357 | 0.1980 | 0.2544 |
| $NGP_{PRUF4PRUF}$ | 0.2442 | 0.7486 | 0.2399 | 0.6540 | 0.2714 | 0.7502 | 0.2550 | 0.2415 |
| $PM_{WORD1DEF}$ | 0.3556 | 0.7433 | 0.3455 | 0.6469 | 0.3344 | 0.7463 | 0.3378 | 0.2425 |
| $PM_{LEM3DEF}$ | 0.2966 | 0.7418 | 0.2654 | 0.6437 | 0.3164 | 0.7482 | 0.3436 | 0.2434 |
| $PM_{POS3DEF}$ | 0.0657 | 0.7299 | 0.0380 | 0.6298 | 0.1118 | 0.7356 | 0.1960 | 0.2542 |
| $PM_{PRUF2DEF}$ | 0.1536 | 0.7442 | 0.1432 | 0.6469 | 0.2114 | 0.7485 | 0.2240 | 0.2415 |
| SPS | 0.2198 | 0.7474 | 0.1962 | 0.6513 | 0.2840 | 0.7537 | 0.2913 | 0.2370 |
| TER | 0.4439 | 0.7521 | 0.4323 | 0.6574 | **0.4643** | 0.7562 | **0.4913** | **0.2324** |
| ALL | 0.4428 | 0.7523 | 0.4541 | 0.6589 | 0.4058 | 0.7557 | 0.4289 | 0.2344 |

Table 2: Automatic evaluation for the range above or equal to 70 percent overlap.

| | $MET_T$ corr | $MET_T$ mean | $NGP_T$ corr | $NGP_T$ mean | $SPS_T$ corr | $SPS_T$ mean | $TER_T$ corr | $TER_T$ mean |
|---|---|---|---|---|---|---|---|---|
| BASELINE | 0.6735 | 0.3372 | 0.5928 | 0.2502 | 0.6667 | 0.3640 | 0.6162 | 0.8687 |
| BEER | 0.6328 | 0.3499* | 0.6031 | 0.2650* | 0.6002 | 0.3650 | 0.4198 | 0.9962 |
| $LEV_{LEM1DEF}$ | 0.6477 | 0.3347 | 0.5559 | 0.2466 | 0.6553 | 0.3643 | 0.6202 | 0.8681 |
| $LEV_{LEM6IGN}$ | 0.6396 | 0.3197 | 0.5905 | 0.2438 | 0.6182 | 0.3493 | 0.5279 | 0.8900 |
| $LEV_{POS4DEF}$ | 0.4930 | 0.3112 | 0.4238 | 0.2309 | 0.5059 | 0.3431 | 0.5032 | 0.8970 |
| $LEV_{PRUF4DEF}$ | 0.5202 | 0.3136 | 0.4676 | 0.2332 | 0.5351 | 0.3435 | 0.4709 | 0.9096 |
| METEOR | 0.6967 | 0.3534* | 0.6723 | 0.2752* | 0.5601 | 0.3531 | 0.3619 | 1.0049 |
| $NGP_{WORD1DEF}$ | 0.7277 | **0.3546*** | 0.7004 | **0.2764*** | 0.6087 | 0.3576 | 0.3587 | 1.0018 |
| $NGP_{LEM4DEF}$ | 0.6732 | 0.3358 | 0.6402 | 0.2614* | 0.5902 | 0.3517 | 0.3935 | 0.9530 |
| $NGP_{POS4DEF}$ | 0.5385 | 0.3158 | 0.4863 | 0.2383 | 0.5067 | 0.3398 | 0.4049 | 0.9453 |
| $NGP_{PRUF4PRUF}$ | 0.5585 | 0.3202 | 0.5334 | 0.2432 | 0.5183 | 0.3412 | 0.3306 | 0.9635 |
| $PM_{WORD1DEF}$ | 0.5577 | 0.3111 | 0.4862 | 0.2204 | 0.5116 | 0.3494 | 0.5956 | 0.7750* |
| $PM_{LEM3DEF}$ | 0.6175 | 0.3220 | 0.5497 | 0.2379 | 0.5869 | 0.3534 | 0.5934 | 0.8259* |
| $PM_{POS3DEF}$ | 0.4872 | 0.3008 | 0.4237 | 0.2149 | 0.4756 | 0.3410 | 0.5540 | 0.8084* |
| $PM_{PRUF2DEF}$ | 0.4738 | 0.3018 | 0.4135 | 0.2142 | 0.4736 | 0.3436 | 0.5358 | 0.7930* |
| SPS | 0.5625 | 0.3196 | 0.4919 | 0.2281 | 0.5804 | 0.3653 | 0.5415 | 0.8095* |
| TER | 0.6533 | 0.3155 | 0.5841 | 0.2235 | 0.6607 | 0.3666 | **0.7075** | **0.7443*** |
| ALL | **0.7663** | 0.3524* | **0.7255** | 0.2687* | **0.7089** | **0.3705*** | 0.6566 | 0.8395* |

Table 3: Automatic evaluation for the range below 70 percent overlap.

## Appendix IV.

### Pearson correlations on the human-evaluated subset

| | HUM corr | $MET_T$ corr | $NGP_T$ corr | $SPS_T$ corr | $TER_T$ corr |
|---|---|---|---|---|---|
| BASELINE | 0.2087 | 0.3637 | 0.3189 | 0.4092 | 0.3731 |
| BEER | **0.2407** | **0.5565** | **0.4836** | **0.5116** | 0.2748 |
| $LEV_{LEM1DEF}$ | **0.2182** | **0.3714** | **0.3220** | **0.4170** | **0.3777** |
| $LEV_{LEM6IGN}$ | **0.2377** | **0.4820** | **0.4963** | **0.4543** | 0.3233 |
| $LEV_{POS4DEF}$ | 0.1816 | 0.2435 | 0.2729 | 0.2676 | 0.2010 |
| $LEV_{PRUF4DEF}$ | 0.1464 | 0.1274 | *0.1416* | 0.1612 | *0.1120* |
| METEOR | 0.1996 | **0.6293** | **0.5579** | **0.4987** | 0.2101 |
| $NGP_{WORD1DEF}$ | **0.2164** | **0.6331** | **0.5946** | **0.5054** | 0.2623 |
| $NGP_{LEM4DEF}$ | **0.2466** | **0.5986** | **0.6043** | **0.4950** | 0.2880 |
| $NGP_{POS4DEF}$ | 0.1938 | 0.3547 | **0.3874** | 0.3106 | 0.1566 |
| $NGP_{PRUF4PRUF}$ | 0.1575 | 0.2732 | 0.2787 | 0.2246 | *0.0937* |
| $PM_{WORD1DEF}$ | **0.3373** | 0.3464 | 0.2681 | 0.3895 | **0.4518** |
| $PM_{LEM3DEF}$ | **0.3100** | **0.5383** | **0.5044** | **0.5221** | **0.4427** |
| $PM_{POS3DEF}$ | **0.2403** | 0.2189 | 0.2121 | 0.2368 | 0.2691 |
| $PM_{PRUF2DEF}$ | **0.2270** | *0.0519* | *0.0297* | *0.1209* | 0.1822 |
| SPS | 0.1955 | 0.2640 | 0.2121 | 0.3403 | 0.2481 |
| TER | *0.1146* | 0.1568 | *0.0899* | 0.3515 | **0.5412** |
| ALL | 0.2407 | 0.3789 | 0.3369 | 0.3383 | 0.2028 |

Table 4: Pearson correlations between the fuzzy matches and the human and automatic evaluation. Results which are higher than the baseline are bolded, statistically insignificant results ($p>0.05$) are in italics.