A Semantic Textual Similarity Enhanced Quality Estimation Method and its Applications to Natural Language Processing Tasks

Hanna Béchara MPhil

A thesis submitted in partial fulfilment of the requirements of the University of Wolverhampton for the degree of Doctor of Philosophy (Ph.D.)

2019

Signature: _____

Date: _____

Abstract

Semantic Textual Similarity (STS) measures the degree of semantic equivalence between two sentences or phrases. Similarity measures between sentences are required in a wide variety of NLP applications, such as information retrieval, topic detection, question answering and automatic text summarisation. Much of the recent work in Machine Learning-based methods for Semantic Textual Similarity takes place within the context of STS tasks and workshops, such as the SemEval workshops, framing the problem as a machine learning task. This thesis primarily investigates the application of semantic textual similarity in evaluation of machine translation (MT) and sets out to answer the following research question: Can semantic textual similarity help accurately predict the quality of MT output? We therefore focus on integrating STS into the machine translation quality estimation (MTQE) process, and propose a novel approach to using STS as a tool to improve evaluation. In order to calculate STS, we also develop several machine learning based methods which are evaluated in SemEval competitions. One which provides a trade-off between the accuracy of the results and its complexity is selected for the experiments reported in the thesis.

Machine Translation Quality Estimation (MTQE) predicts the quality of machine translation output without the need for a reference translation. This quality can be defined differently based on the task at hand, be it post-editing, quality assurance, or system ranking. In an attempt to focus further on the adequacy and informativeness of translations, we integrate features of semantic similarity into QuEst, a framework for MTQE feature extraction. By using methods previously

employed in Semantic Textual Similarity (STS) tasks, we use semantically similar sentences and their quality scores as features to estimate the quality of machine translated sentences. Our experiments show that finding semantically similar sentences for some datasets is difficult and time-consuming. Therefore, we opt to start from the assumption that we already have access to semantically similar sentences. We test our hypothesis on three different datasets, including one of our own design. Our results show that this method can improve the prediction of machine translation quality for semantically similar sentences.

Furthermore, this thesis poses the research question: To what extent does the use of quality estimation tools affect the efficiency of the translation workflow? To test our technique in a real-world setting, we design a user study engaging professional translators in post-editing tasks using MTQE. To assess the translators' cognitive load we measure their productivity both in terms of time and effort (keystrokes) in 3 different scenarios: translating from scratch, post-editing without using MTQE, and post-editing using MTQE. We also investigate the impact of accurate MTQE versus inaccurate MTQE. We conduct our user study with 4 professional English to Spanish translators, using a modified version of PET (Post-Editing Tool[1]) as our post-editing tool. Our results show that good MTQE information can improve post-editing efficiency and decrease the cognitive load on translators. We conclude that MTQE can be an effective tool to pre-emptively assess the quality of MT systems to avoid underpayments and mistrust by professional translators.

Finally, we explore the impact of STS in other fields of evaluation. We ask a further research question: Can we expand the applications of Semantic Textual Similarity for evaluating the quality of automatically simplified text? What about translation memory matching? In order to answer these questions, we apply the same STS techniques to evaluate the output of automatically simplified text. As text simplification is monolingual, we can directly apply the STS tools to the original

[1] `http://www.clg.wlv.ac.uk/projects/PET/`

and simplified sentence pairs. We augment our system with features that detect fluency and simplicity. We find that our features are strong indicators for quality, especially in preserving meaning after simplification. On the Shared Task on Quality Assessment for Text Simplification (QATS), our classification systems ranked second overall among all participating systems and consistently outperformed the baseline for all types of quality measures. In the case of translation memory matching, we find that STS-enhanced methods perform comparatively with Edit Distance (ED) methods, which remain the most widely used methods for matching and retrieval today. We conclude that STS-based retrieval can be useful in cases where ED cannot find a match. However, due to its simplicity and efficiency, ED remains the better choice for TMs.

Contents

List of Figures

List of Tables

Acknowledgements

This thesis would not have been possible without the hard work and support of the following people:

First and foremost, my supervisor and director of studies, Dr Constantin Orăsan, whose guidance, supervision and good humour were invaluable, and pushed me to do more than I thought I was capable. Second, to my supervisor Dr Marcos Zampieri, who despite being a little late to the party was still vital in its process. And third, to Dr Carla Parra, and her insistence that there is life after the PhD thesis. What else can I say, this work would not have been possible without them.

Furthermore, my sincerest thanks go out to my collaborators and partners in crime: To Dr Sanja Štajner, and those late nights running experiments in the lab or playing board games on Evans Street. To Dr Rohit Gupta, and weekends spent navigating the streets of Madrid, Rome and Sheffield. And to all my friends and co-workers at RGCL, especially: Dr Shiva Taslimipoor, Dr Victoria Yaneva, Dr Sara Moze, Dr Michael Oaks and (soon to be) Dr Richard Evans.

I would also like to extend a sincere thank you to Dr Lucia Specia and Dr Carolina Scarton who made my freezing secondment in Sheffield just a little bit warmer.

Finally, I would like to thank the professors who set me on this path: Prof Josef van Genabith, Prof Detmar Meurers and Prof Joseph Béchara.

Chapter 1

Introduction

Since the introduction of statistical machine translation (SMT) in 1990 (Brown et al., 1990), the field has evolved quickly with MT research evolving from rule-based models to example based models, statistical models, hybrid models, and more recently neural models (Han and Wong, 2016). In the context of these advances in machine translation tools, comparative assessment of the various outputs is a challenging yet important part of the process. Developers have turned to a variety of techniques to assess the quality of machine translation output. While many consider human evaluation to be the best and most reliable judgement in machine translation evaluation, this method is inefficient and expensive, especially when large corpora are involved (Bojar et al., 2017). Automatic evaluation metrics have been developed to estimate MT output quality, but these rely on reference translations and focus mainly on syntactic and surface similarities, rather than semantic accuracy. Therefore, machine translation quality estimation (MTQE) and machine learning techniques have become one of the focuses of MT output evaluation, as they can be used to measure different aspects of correctness. One aspect of correctness that has not been subject of enough research is the notion of semantic correctness. While several tools that measure monolingual similarity have been developed, the extent to which such tools can help in machine translation evaluation across languages has

not been fully researched.

This thesis focuses mainly on addressing the question of informativeness and semantic soundness in machine translation evaluation. Semantic textual similarity (STS) measures the degree to which two sentences are semantically equivalent (Agirre et al., 2012). Similarity measures between sentences are required in a wide variety of NLP applications, such as information retrieval (Bhatia et al., 2013), topic detection, question answering (Mohler et al., 2011) and automatic text summarisation (Aliguliyev, 2009). Much of the recent work in Machine Learning-based methods for Semantic Textual Similarity takes place within the context of STS tasks and workshops, such as the SemEval workshops (Agirre et al., 2012, 2013, 2015, 2016), framing the problem as a machine learning task. We make and support the claim that STS can help determine the degree to which information is preserved after translation, and the degree to which it is lost. In order to explore that claim, we start by developing several machine learning based methods which are evaluated in the SemEval competitions. We then propose a novel approach to integrating STS into the MTQE pipeline, through the use of semantically similar sentences with quality scores. We test our hypothesis on three different datasets, including a dataset which we ourselves designed to fit the purpose. Our results show that this method can improve the prediction of machine translation quality for semantically similar sentences. We also test our method in a real-world setting, using professional translators to act as post-editors and to evaluate the impact that our STS enhanced MTQE method has on post-editing efficiency. We conclude that MTQE can be an effective tool to preemptively assess the quality of MT systems. We also extend the application of the STS tool to the evaluation of automatically simplified text and to the retrieval of translation memory matching. It shows competitive results in the evaluation of automatic text simplification, although it does not outperform basic Edit Distance for translation memory retrieval.

1.1 Research Questions

Our research focuses on investigating the following research questions:

RQ1 Can semantic textual similarity help accurately predict the quality of MT output?

RQ2 To what extent does the use of quality estimation tools affect the efficiency of the translation workflow?

RQ3 Can we expand the applications of Semantic Textual Similarity further:

RQ3.1 in automatic evaluation of simplified text?

RQ3.2 in translation memory matching and retrieval?

RQ1: Semantic Textual Similarity with MTQE

Our first research question explores whether and to what extent semantic data can be used to improve the MT evaluation process.

RQ1 Can semantic textual similarity help accurately predict the quality of MT output?

This is our first and most general research question. Previous work on semantic textual similarity (STS) has focused mainly on monolingual similarity. That is, it is designed to work best between similar sentences of the same language, rather than across languages. With that in mind, we take an indirect approach to incorporating STS into the evaluation process. We attempt to compare our MT output to similar sentences that themselves may have a reference translation or a predetermined evaluation rating. By incorporating this information into the quality estimation pipeline as features, we investigate whether or not STS can be used to determine

how semantically sound a translation can be. This method introduces several challenges, as current STS systems can be slow and inaccurate, and searching a vast corpus of sentences in order to find one that is similar enough to matter can be time-consuming.

RQ2: MTQE in a Real World Setting

Our second research question concerns the real-world application of such evaluation methods. One of the main theoretical real-world applications of reference-free evaluation is that it can be used as part of the computer assisted machine translation workflow to speed up the post-editing process. Translators who use machine translation as an aid during the translation process need to decide whether a sentence is worth post-editing or is faster translated from scratch. In theory, a good quality estimation tool could make that decision for the translator and increase their efficiency. However, the extent to which MTQE tools are used in real world translation settings remains limited. Therefore, we pose the following research question:

RQ2 To what extent does the use of quality estimation tools affect the efficiency of the translation workflow?

In order to answer this research question, we design a user study to test the effect of MTQE on post-editors in a professional setting. we design a traffic light system to present translators with three different categories of sentences and determine how effective MTQE is at improving the efficiency of the translation workflow.

RQ3: Other Applications of STS as a Tool for Evaluation

Text simplification is the process of transforming a text into another, easier to read and understand text, while still preserving the same information (Shardlow, 2014).

This transformation involves the replacement of complex sentences into shorter and simpler ones, and the lexical replacement of difficult words with their simpler synonyms. The primary objective of text simplification is to make complex texts more accessible to a wider variety of audiences, such as those with learning disabilities. Automatic text simplification is a large field of research that spans 20 years, and seeks to automate this process, making it easier to convert large corpora into a more readable version. The monolingual nature of automatic text simplification and the importance of semantic preservation makes STS a logical tool to use for its evaluation. Therefore, we pose the following the research question:

RQ3 Can we expand the applications of Semantic Textual Similarity further:

RQ3.1 in automatic evaluation of simplified text?

RQ3.2 in translation memory matching and retrieval?

1.2 Roadmap

This thesis takes on the task to answer these questions. We provide the necessary background work that frames the context of our own experiments and contributions. The following paragraphs describe the organisation and structure of this report.

Chapter 2 presents a review of the relevant literature and frames our motivation for this study. We first introduce the concept of STS and its developement from recognising textual entailment (RTE). We look at a few relevant tools for STS and provide a snapshot of the history of RTE and STS workshops. We further outline the evolution of both reference-based and reference-free machine translation evaluation tools, including the tools used in this research.

Chapter 3 describes the STS systems developed over the course of this research and the features and tools used in building this system. In this chapter we present

two systems, submitted to SemEval 2014 and 2015. These systems build on the research presented in Chapter 2. The first system consists of a total of 31 features extracted from pre-existing language technology tools, a paraphrasing database, machine translation evaluation tools and corpus pattern analysis. We then use machine learning tools to build a supervised regression model to predict the similarity between two sentences. The 2015 system builds on this work, adding a number of distributional, conceptual and semantic similarity measures, along with features based on multiword expressions. Both systems outperformed the baseline system used by the workshop organisers. The final STS system used in this research is a streamlined version of these two submissions, optimised for efficiency and performance after extensive feature selection, and is comprised of 13 language technology and paraphrasing features.

Chapter 4 looks at ways to incorporate what we developed in Chapter 3 into the evaluation process. In this chapter we present a novel approach that uses a previously evaluated and semantically similar sentence, in order to determine the quality of a machine translated sentence. We test our method in a series of experiments on different datasets that investigates the effectiveness of our approach. We also describe the design of a new dataset, consisting of semantically similar sentences and their machine translations, with manual evaluations of the translations. While small, this dataset provides a solid medium in which to test our approach. Overall, our results showed a consistent improvement in accuracy over a baseline quality estimation system.

Chapter 5 details the user study we designed to test the impact of our evaluation system in a real-world setting. The study enlists the work of 4 professional translators and compares their efforts to post-edit (PE) both with and without MTQE. It also compares the impact of the use of accurate MTQE to inaccurate MTQE. For this study, we assembled a dataset of 260 English sentences, their Spanish MT translations, and their post-edited translations. We divided the sentences into 4 dif-

ferent categories: sentences without MTQE, sentences with good/accurate MTQE, sentences with bad/inaccurate MTQE, and sentences to be translated without PE. The sentences were presented to the translators using a traffic light system with green telling the user to post-edit, red telling the user to translate from scratch and blue telling them to make up their own mind. Our results showed that good MTQE has a positive impact on the efficiency of the translation workflow, and can cut translating time and effort significantly.

Chapter 6 explores the possibility of further applications of our STS approach to evaluation. We test our method both in evaluating automatic text simplification and translation memory matching, due to the monolingual nature of both these tasks. We apply the same STS techniques to evaluate the output of automatically simplified text. As text simplification is monolingual, we can directly apply the STS tools to the original and simplified sentence pairs. However, as semantic similarity is only one of many aspects of ATS evaluation, we augment our system with features that detect fluency and simplicity. We find that our features are strong indicators of quality, especially in preserving meaning after simplification. On the Shared Task on Quality Assessment for Text Simplification (QATS), our classification systems ranked second overall among all participating systems and consistently outperformed the baseline for all types of quality measures. In the case of translation memory matching, we find that STS-enhanced methods perform comparatively with Edit Distance (ED) methods, which remain the most widely used methods for matching and retrieval today. We conclude that STS-based retrieval can be useful in cases where ED cannot find a match. However, due to its simplicity and efficiency, ED remains the better choice for TMs.

And finally, Chapter 7 sums up our research to date, outlining the main conclusions and outcomes, in addition to avenues for future research.

1.3 Resources and Publications

We present two separate datasets which we created throughout our research:

- FLICKR EN-FR Dataset[2]: A dataset of semantically similar English sentence pairs, with an STS score and their manually evaluated French machine translations.
- Autodesk Real-World Data[3]: A subset of 260 sentences from Autodesk post-editing data, divided by accuracy of MTQE.

Several papers were published in peer-reviewed conference proceedings and workshops as part of the research presented in this thesis. Our work on STS systems appeared in publications for the SemEval workshop and were presented as posters in 2014 (7) and 2015 (6) at the workshops. Our work on integrating STS into the MTQE pipeline was published in the Proceedings of the 19th Annual Conference of the European Association for Machine Translation (EAMT) (3). The preliminary findings on our user study were published in the 2017 edition of The Prague Bulletin of Mathematical Linguistics (1). The full study is currently under review for the special issue *Advances in Computer-Aided Translation Technology*, and at the time of this writing, awaits reviewer feedback.

A full list of publications from this thesis:

(1) Parra Escartín, C., Béchara, H. and Orăsan, C., 2017. **Questing for Quality Estimation: A User Study.** The Prague Bulletin of Mathematical Linguistics, 108(1), pp.343-354.

This paper published the preliminary findings for the User Study presented in Chapter 5. The study itself was designed by all 3 co-authors. The original data was ex-

[2] `https://gist.github.com/hbechara/333c8be2d03515b6b6cc39b4deeeaffc`
[3] `http://dinel.org.uk/projects/postediting-dataset/`

tracted and analysed by H. Béchara, and further analysis was provided by the other two authors. The first author, C. Parra Escartín, was in charge of the preparation of the paper, putting together the text she received from the other authors and her contribution. The results section and their analysis was written by H. Béchara, who also carried out the data analysis.

(2) Štajner, S., Popovic, M. and Béchara, H., 2016. **Quality Estimation for Text Simplification.** In Proceedings of the LREC Workshop on Quality Assessment for Text Simplification, pp. 15-21.

This paper presents an early version of using QE for the Assessment for Text Simplification. The method itself was suggested by the first author S Štajner. The engineering of the features used in the paper was the joint work of the three authors. The paper was written by the first author, S Štajner, with contributions from the other two authors.

(3) Béchara, H., Parra Escartín, C., Orăsan, C. and Specia, L., 2016. **Semantic Textual Similarity in Quality Estimation.** In Proceedings of the 19th Annual Conference of the European Association for Machine Translation (pp. 256-268).

This paper presents the method for integrating STS features into MTQE, laid out in Chapter 4. The method itself was designed by H. Béchara, and C. Orăsan and implemented by H. Béchara. The paper describes three different sets of experiments, designed and carried out by H. Béchara with feedback and suggestions from all three co-authors. The paper itself was written by H. Béchara with contributions and feedback from all three co-authors.

(4) Béchara, H., Gupta, R., Tan, L., Orăsan, C., Mitkov, R. and van Genabith,

J., 2016. **Wolvesaar at SemEval-2016 task 1: Replicating the success of monolingual word alignment and neural embeddings for semantic textual similarity.** In Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016) (pp. 634-639).

This paper was a joint contribution between the University of Wolverhampton and the University of Saarland and describes our submission to the SemEval 2016 workshop. The system was designed by the first three authors (Béchara, Gupta and Tan), who also jointly wrote the paper with feedback and input from the rest of the co-authors.

(5) Štajner, S., Béchara, H. and Saggion, H., 2015. **A Deeper Exploration of the Standard PB-SMT Approach to Text Simplification and its Evaluation.** In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers) (Vol. 2, pp. 823-828).

This paper describes a series of experiments that examine the result of phrase-based machine translation tools applied to Text Simplification. It also uses machine translation metrics to evaluate the output of these tools. The method and experiments were suggested by the first author (Stajner) and designed and run by the second author (Béchara). The results were analysed and written up by the first author, with contributions and feedback from the second and third authors.

(6) Béchara, H., Costa, H., Taslimipoor, S., Gupta, R., Orăsan, C., Pastor, G.C. and Mitkov, R., 2015.**Miniexperts: An SVM Approach for Measuring Semantic Textual Similarity.** In Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015) (pp. 96-101)

This paper was our second submission to the SemEval workshop and built on the system we designed in 2014. The system was designed by the first two authors (Béchara and Costa) but built on previous work by the fourth author (Gupta). The third author (Taslimipoor) contributed one of the features used in the model. The paper was written by the first and second author (Béchara and Costa) with contributions and feedback by the rest of the co-authors.

(7) Gupta, R., Béchara, H., El Maarouf, I. and Orăsan, C., 2014.**UoW: NLP techniques developed at the University of Wolverhampton for Semantic Similarity and Textual Entailment.** In Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014) (pp. 785-789).

This paper describes our first submission to the SemEval workshop in 2014. The model was built on features chosen and tested by the first two authors (Gupta and Béchara) along with El Maarouf who contributed the CPA features. The models were trained and run by the first two authors and the paper was written jointly between them along with contributions and feedback from all the co-authors.

(8) Gupta, R., Béchara, H. and Orăsan, C., 2014. **Intelligent translation memory matching and retrieval metric exploiting linguistic technology**. Proceedings of Translating and the Computer, 36, pp.86-89.

This paper presents some preliminary findings from experiments designed to test the use of STS form TM matching and retrieval. The experiment was designed and tested by the first author (Gupta). The second author (Béchara) carried out the manual evaluation and wrote the part reporting this analysis. The paper itself was written by the first author (Gupta) with contributions and feedback from all the co-authors.

Chapter 2

Background Information

The purpose of this chapter is to provide the background information required for understanding the thesis and provide an overview of the methods that are used throughout the thesis. In addition, it provides some historical overview of how the topics covered in this thesis developed. We will give an overview of the state of the art in the relevant fields of research and provide context for this thesis. The literature review is divided as follows: Semantic Textual Similarity (STS) (Section 2.2) and Machine Translation Evaluation (Section 2.3) . Interest in the field of STS developed partially as a result of the successes obtained in the field of Recognising Textual Entailment (RTE). For this reason, Section 2.1 gives an overview of RTE and how research into it evolved into STS. The section on Machine Translation Evaluation covers both human and automatic evaluation systems, including metrics and quality estimation. It also provides an overview of the Workshop for Machine Translation (WMT) shared tasks on Quality Estimation, where much of the state of the art and the most important frameworks have been developed.

The rest of this chapter is organised as follows. As research into semantic textual similarity is inextricably linked to textual entailment, Section 2.1 provides an overview of recognising textual entailment (RTE) and its tasks and challenges. Sec-

tion 2.2 delves into the history and state of the art of semantic textual similarity, with a focus on the SemEval shared tasks that provided a medium for most of the research in this field. Section 2.3 lays out a state of the art of machine translation evaluation, covering both human and automatic evaluation methods, and discussing both reference-free and reference-based methods of automatic evaluation. Chapter 2.4 provides some motivation for our work and the design choices made throughout the thesis. The Chapter ends with our conclusions in Section 2.5.

2.1 Recognising Textual Entailment

Research in STS is closely tied to research into Recognising Textual Entailment (RTE) tasks (Dagan et al., 2010). Therefore, to glean a full understanding of the history of STS, we have to begin with RTE.

RTE recognises whether the meaning of a text can be inferred (entailed) from another. In other words, a text A is said to entail a text B when the meaning of B can be inferred from the meaning of A. This means that entailment is a unidirectional relation. As discussed later, this differs from STS, which is bidirectional. Example (1), taken from Sammons et al. (2011), demonstrates a text and three hypothetical entailments.

(1) The purchase of Houston-based LexCorp by BMI for $2 Billion prompted widespread sell-offs by traders as they sought to minimize exposure. LexCorp had been an employee-owned concern since 2008.

   a. Hyp 1: BMI acquired an American company.

   b. Hyp 2: BMI bought employee-owned LexCorp for $2 Billion

   c. Hyp 3: BMI is an employee-owned concern

De Marneffe et al. (2008) expand the definition of entailment to include a third op-

tion: contradiction. A sentence "A" contradicts a sentence "B" when both sentences are highly unlikely to be true at the same time. The authors introduce a three-way classification system that labels sentences as either *Entailed*, *Contradicted* or *Unknown*. In Example (2), Hyp 2 contradicts the original text, as LexCorp could not have been bought by BMI and remain employee-owned. De Marneffe et al. (2008) posit that while contradictions can arise from obvious features such as antonymy, negation and numeric mismatches, they can also arise from more complex structures and are therefore more difficult to determine than entailment. This is demonstrated in the example in Example (2) below, where Hyp 1 neither follows nor necessarily contradicts the text, as the 100 people could still have been injured regardless of the rocket's defusing.

(2) Police specializing in explosives defused the rockets. Some 100 people were working inside the plant.

a. Hyp 1: 100 people were injured.

The task of RTE is a less complex problem than that of semantic similarity, yet its appeal remains broad due to its uses in various other NLP applications from Automatic Question Answering (Harabagiu and Hickl, 2006), to Machine Translation (Mirkin et al., 2009) and Information Retrieval (Clinchant et al., 2006). In automatic question answering, RTE can be used to rerank sentences or eliminate sentences that do not meet the basic requirements of entailment. Harabagiu and Hickl (2006) show that such methods increase the accuracy of question-answering by as much as 20%. In SMT, RTE has been applied to solve the problem of unknown words, where the SMT system fails to translate a word it has not previously encountered. Mirkin et al. (2009) address this by paraphrasing the sentences prior to translation, using both a paraphrase database and entailed texts to generate multiple alternatives to the original sentence. A manual evaluation showed a strong preference for the system that used entailment for paraphrasing, by human evaluations. RTE models have

also been used in Information Retrieval, where the entailment can be used to capture dependencies between query and documents which are not captured by simple word-based similarities.

2.1.1 A History of the RTE Challenge

Research in the field of textual entailment benefited greatly from the RTE challenge. Dagan et al. (2005) introduced the RTE challenge as an application-independent task that asks participants to label a pair of sentences with *True* if one sentence entails the other and *False* if it does not. The RTE task dedicated itself to the problem of RTE, setting the baseline and benchmark for RTE systems and providing common grounds for researchers to share and compare their work. The shared task provided the participants with an annotated dataset of small text snippets from the news domain. The dataset was collected and then labelled by human annotators, within different application settings such as Information Retrieval, Question Answering, Paraphrase Acquisition and others. The human annotators selected 50% of the sentence pairs for entailment and the other 50% for contradiction. Examples (3) and (4) below are taken from the data description in Dagan et al. (2005).

(3) Question Answering Subtask

a. Text: The Republic of Yemen is an Arab, Islamic and independent sovereign state whose integrity is inviolable, and no part of which may be ceded

b. Hypothesis: The national language of Yemen is Arabic.

c. Value: True

(4) Information Extraction Subtask

a. Text: Regan attended a ceremony in Washington to commemorate the landings in Normandy.

b. Hypothesis: Washington is located Normandy.

c. Value: False

Participants were provided with a training set and a development set for tuning. Participants were to treat the task as a classification problem and provide an automatic label (and an optional confidence score) for each sentence pair. Sixteen systems were submitted that covered a large range of approaches, ranging from very basic word-overlap systems to statistical methods. Overall, the system accuracies were between 60 and 70 percent, and made the first step into identifying textual entailment as its own discipline with NLP.

Since then there have been promising improvements in RTE with different researchers providing different approaches to the problem. The following year, the challenge provided a larger and more "realistic" dataset, with 800 sentence pairs for developement and another 800 for testing (Bar-Haim et al., 2006). It also added a subtask, which asked the participants to rank the entailed sentences by confidence. This challenge saw more participants, with 23 submissions and results ranging between 53% and 75% accuracy. The highest performing system uses a classication-based approach to combine lexico-semantic information derived from text processing applications with a large collection of paraphrases acquired automatically from the web (Hickl et al., 2006). Bos and Markert (2006) presented their contribution to the RTE task, comparing and combining a shallow semantic method with a method based on logical inference. The first method used a bag-of-words approach to measure word overlap. The deeper semantic analysis made use of two kinds of automatic reasoning tools: first order theorem proving and finite model building, resulting in a set of features that they hoped would show where a sentence "A" entails a sentence "B". At first glance, adding the deeper analysis did not improve over shallow features. The system achieved a best run of 60.1% accuracy.

The third RTE challenge followed closely in the footsteps of the 2006 challenge,

but introduced longer texts into the dataset (Giampiccolo et al., 2007). Most notably, this task piloted a third option, **UNKNOWN**, to the entailment task, for situations where a hypothesis sentence neither entails nor contradicts the text. 26 teams participated, with the overall accuracies of systems between 35% and 73%. Once again, the best performing system, (Hickl and Bensley, 2007), used a statistical model for commitment extraction, lexical alignment, and entailment classification.

The fourth challenge incorporated the previous pilot task into its main task, now using a three-way classification system (**ENTAILS, CONTRADICTS, UNKNOWN**) in the main task (Giampiccolo et al., 2008). However, a second two-way task was also created by automatically converting the labels **CONTRADICTS** and **UNKNOWN** into **NO ENTAILMENT**. Twenty-six teams participated in that year's challenge. The average accuracies of the systems were 58% for the two-way system and 30.7% for the three-way system. The highest reported accuracy was 74.6% for the two-way system.

The fifth challenge added a new task, the search task, which consisted of searching a corpus for all the sentences that entailed a given text (Bentivogli et al., 2009). This new task became the focus of the last two RTE challenges, shifting from recognising textual entailment to retrieving entailing sentences. This was the last challenge dedicated specifically to RTE. However, the classical RTE recognition task continued with other workshops, such as the SemEval workshops (further described in Section 2.2.3) which often included a Textual Entailment (TE) component or subtask.

2.2 Semantic Textual Similarity

STS is defined by Agirre et al. (2012) on page 1 as a "measure that captures the notion that some texts are more similar than others". STS measures this degree of

semantic similarity which can range from completely unrelated to exact semantic equivalence. The concept of STS builds on RTE in the sense that both measure and compare texts on a semantic level. While TE only needs to hold true in one direction, STS assumes the similarity relationship between the two snippets of text is bi-directional. Furthermore, TE is binary: A sentence can either entail another, or not. On the other hand, STS can be graded on a scale, depending on how much the texts differ.

STS has applications in several fields of research. One such application is in the field of text summarisation. Text summarisation is the automatic creation of shorter texts from longer texts or a collection of longer texts, while still preserving the meaning of the original text. Graph-based summarisation relies on similarity measures in its edge weighting mechanism (Aliguliyev, 2009). Information retrieval (IR) is the process of extracting related information from large collections of information resources, and presenting it according to a user's need (Bhatia et al., 2013). In IR, similarity measures are used to assign a ranking score between a query and texts in a corpus. Question Answering (QA) is a specific type of information retrieval where a system tries to find the correct answer to a given question. Question answering applications require similarity identification between a question-answer or question-question pair (Mohler et al., 2011).

2.2.1 Semantic Similarity in Human Translation Studies

Semantic similarity is a well-studied and complex problem in human translation studies. Human linguists identify two main fields of semantics: logical semantics, which deal with sense, reference and implication, and lexical semantics, concerned with word and phrase meanings, and the relationship between them. Our research focuses mostly on determining lexical semantics, both on a word and phrase level. This is itself not a simple task, as there is no one-to-one relationship between or-

thographic words and elements of meaning (Baker, 2011). Furthermore, according to Cruse et al. (1986), there are four main types of meanings in words:

- Propositional Meaning: The relation between a word and what it represents in the world. This is the type of meaning that we use when we speak of equivalence and correctness.
- Expressive Meaning: Relates to feeling and attitude around words and phrases. We do not take this meaning into account when measuring semantic similarity, as it is highly contextual and cannot be judged as true or false.
- Presupposed Meaning: Relies on co-occurrence restrictions and deals with expected meaning and collocational restrictions. This includes figurative language, idioms and restrictions that are highly language-dependent.
- Evoked Meaning: Which arises from dialect and register, and can fall into geographical, temporal or social categories.

In particular, we are mostly interested in propositional semantics as the most straightforward measure of semantics between written texts, and the only one that really deals with an objective semantic value. Therefore, when we speak of similarity and correctness in this thesis, we generally are referring to propositional similarity and correctness. As we concern ourselves with sentence-level semantic similarity, we are interested in word-level and phrase-level. Therefore, we do not deal with text cohesion and coherence, pragmatic similarity, ethics or other larger picture concepts of similarity. While these are important aspects of correctness and similarity, they are beyond the scope of this research.

The automatic methods to determine semantic similarity used in computational linguistics (and by extension this research) are very often highly data driven. Therefore, research in computational linguistics quite often fails to address theories proposed by linguists. For the most part, the methods are quite shallow by comparison

and focus on learning from data rather than theory. For this reason this discussion is beyond the scope of this thesis.

2.2.2 Early Research into Semantic Textual Similarity

One of the earlier methods for measuring semantic textual similarity is presented by Mihalcea et al. (2006), notable in that it was the first of its kind to focus on short texts rather than individual words or full documents. Their approach used a combination of corpus-based and knowledge-based measures. Corpus-based measures attempt to model the similarity between words by using information from very large corpora and measuring co-occurrence to determine a dependency between two words. Knowledge-based measures, on the other hand, use semantic networks to quantify the degree to which two words are related. The authors test the effectiveness of their method on paraphrased sentences from the Microsoft paraphrase corpus (Dolan et al., 2004). They used 4,076 sentences for training and 1,725 for testing. They report an overall final accuracy of 70.3%, compared to an 83% accuracy when using human evaluators. Overall, the system improved dramatically over traditional vector-based methods.

Another early foray into semantic textual similarity for short texts comes from Gabrilovich and Markovitch (2007), who use machine learning techniques to represent the meaning of any text as a weighted vector of Wikipedia-based concepts. They call their approach ESA (Explicit Semantic Analysis). In their approach, texts are represented by weighted vectors of concepts that are compared using the cosine metric. This is similar to the work presented by Mihalcea et al. (2006) in that both approaches manipulate a collection of concepts. However, where the previous approach compares the degree of relatedness between words, this approach treats words and texts the same way, leading to a higher degree of word sense disambiguation as words always appear in context. Their results show vast improvements over previous

methods, with 0.75 correlation with human judgements, slightly outperforming the system previously proposed by Mihalcea et al. (2006).

2.2.3 The SemEval Shared Tasks

Much of the recent work in Machine Learning-based methods for Semantic Textual Similarity takes place within the context of STS tasks and workshops, such the SemEval workshop, framing the problem as a machine learning task. SemEval's shared tasks have been particularly interested in semantic similarity, working to fine-tune and perfect these similarity measures, and explore the nature of meaning in language. These models compare a pair of texts and provide a score that measures similarity based on a scale from 1-5, as defined in Table 2.1.

What follows is a short overview of each of the SemEval workshops involving STS since its inception, and the best systems in each workshop.

SemEval-2012 Task 6: A Pilot on Semantic Textual Similarity

A shared task for determining semantic textual similarity first took place at the workshop for semantic evaluation (SemEval) in 2012, as SemEval-2012 Task 6: A Pilot on Semantic Textual Similarity.

The instructions for the task, given by Agirre et al. (2012), were as follows: *Given two sentences, s1 and s2, participants will quantifiably inform us on how similar s1 and s2 are, resulting in a similarity score. Participants will also provide a confidence score indicating their confidence level for the result returned for each pair. Participants will be asked to explicitly characterize why a pair is considered similar, i.e. which semantic component(s) contributed to the similarity score.*

For this task, the organisers assembled 3 datasets from different sources.

Table 2.1: Semantic Textual Similarity scale used by SemEval

1	**The two sentences are on different topics** "A man is jumping into an empty pool." "There is no biker jumping in the air."
2	**The two sentences are not equivalent, but share some details** "Two children are lying in the snow and are making snow angels." "Two angels are making snow on the lying children."
3	**The two sentences are roughly equivalent, but some important information differs/is missing** "The young boys are playing outdoors and the man is smiling nearby." "There is no boy playing outdoors and there is no man smiling"
4	**The two sentences are mostly equivalent, but some unimportant differs/missing** "Four girls are doing backbends and playing in the garden" "Four girls are doing backbends and playing outdoors"
5	**The two sentences are completely equivalent** "The current is being ridden by a group of friends in a raft." "A group of friends are riding the current in a raft"

- MSRPar: Sampled from the Microsoft Research Paraphrase Database (Dolan et al., 2004) at certain ranks of string similarity. 1500 sentences overall (divided into 50% for training and 50% for testing)

- MSVid: Sampled from the MSR Video Paraphrase Corpus (Chen and Dolan, 2011), a corpus assembled by showing annotators videos and asking them to write a short sentence describing the video. 1500 sentences overall (divided into 50% for training and 50% for testing)

- Pairs from the translation shared task of the 2007 and 2008 ACL Workshops on Statistical Machine Translation (WMT) (Callison-Burch et al., 2007, 2008)

matched with their machine translations. 1830 sentences in total from different domains.

In total, 35 teams participated and submitted 88 runs. The output of the participant systems was compared to manual scores, using a Pearson Correlation Coefficient and Mean Square Error. The baseline to beat was a simple word overlap system. The Pearson Correlation Coefficient is defined as the ratio of the covariance of two variables representing a set of numerical data, normalised to the square root of their variances, as shown in Equation 2.1.

$$P_{XY} = \frac{\mathbf{Cov}(X,Y)}{\sqrt{\mathbf{Var}(X)\mathbf{Var}(Y)}} \tag{2.1}$$

Mean Squared Error (MSE) is calculated as the mean of the squares of difference between the predicted and observed results, as demonstrated in Equation 2.2.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(X_i - Y_i)^2 \tag{2.2}$$

where Y is the vector of predictions and X is the vector of observed values.

The top performing systems in this task achieved a mean correlation of 0.677 and 0.675 respectively. The best system, submitted by Bär et al. (2012) uses a log-linear regression model, combining various text similarity measures ranging from simple n-grams matches to complex Explicit Semantic Analysis vector comparisons and aggregation of word similarity based on lexical-semantic resources. This system achieved an overall correlation of 0.677, versus a baseline of 0.31.

TakeLab[4], the second best system submitted by Šarić et al. (2012), follows a set-up similar to that described in Bär et al. (2012). In contrast, the authors use a

[4]`http://takelab.fer.hr/sts/`

support vector regression model to predict semantic similarity, and employ multiple features measuring word overlap and syntax similarity. This system performed particularly well on paraphrasing datasets, and achieved a correlation of 0.675. TakeLab is still used as a baseline for many STS tasks today.

In their submission to the same workshop, Jimenez et al. (2012) present an approach to text similarity using what they call "soft cardinality". Soft cardinality takes into account commonalities and differences between sets and weighs them accordingly. By using a set-based weighted soft-similarity method, they compare sentences, words and characters using surface information. They ranked third among 89 systems, achieving a score of 0.6071. The overall average Pearson score for all systems was 0.56.

The 2012 workshop served as a pilot and their submissions set the stage for future tasks as well as the benchmark for further systems to beat. Future workshops would build on the data and evaluation system used in this task.

SemEval 2013 Task 6: Semantic Textual Similarity

Following the success of the pilot study, the STS shared task continued the following year at SemEval2013, also as Task 6: Semantic Textual Similarity (Agirre et al., 2013). The task description was unchanged, and the same datasets from 2012 were used but augmented with newswire headlines and additional MT sentences. That year also introduced a second subtask that attempted to ask why two sentences were deemed similar. The total number of teams that year was 34, with 89 runs submitted overall. The baseline ranked at 73 out of 89 runs, with a mean correlation of 0.364.

The best performance was the submission by Han et al. (2013), who described a new feature for semantic similarity based on distributional similarity and WordNet

path similarity. Their method used WordNet synsets and hypernyms to find relationships between words. They submitted 3 separate runs, which ranked first, second and fourth, with Pearson scores of 0.618, 0.593 and 0.568 respectively. Wu et al. (2013) explored three semantic representations: named entities, semantic vectors, and structured vectorial semantics, and combined them with then-current state-of-the-art features. Using a feature-selection algorithm, they chose the most effective features and improved the existing systems. Their system scored fifth, sixth and eighth among the 89 runs, with Pearson scores of 0.567, 0.564 and 0.557 respectively.

SemEval 2014 Task 1: Evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment

In SemEval 2014, the STS shared task returned under the description: "Evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment" (Marelli et al., 2014a). This task included two subtasks, one calling for semantic similarity and one calling for textual entailment. This is the first time that textual entailment was included in the STS task.

This time organisers used the SICK dataset (Marelli et al., 2014b), annotated for both similarity and entailment. More details about this dataset are presented in Section 3.2.1. Sixteen teams participated in both subtasks. On the STS subtask, the systems scored between 0.47 and TE subtask, the accuracy was between 48% and 84%. The organisers drew a distinction between compositional and non-compositional features for this task, and found that compositional features' performance were comparable with the average results obtained in the task.

The top performing system, further detailed in Zhao et al. (2014), attempted to solve both tasks at once by treating STS as a regression problem and RTE

as a classification problem, and using the same set of features for both problems. They extracted seven different categories of features: features focusing on the length of segments, surface similarity features, semantic similarity features that captured contextual data, grammatical dependencies, string features (such as n-gram matchings and world overlap), text difference measures, which looked at negations and antonyms, and corpus-based features (which looked at co-occurrences). This system achieved a mean Pearson score of 0.828. A close second, with a Pearson score of 0.827, was the system presented in Bjerva et al. (2014). This system also took a supervised approach with a variety of features, ranging from simple word-overlap, to more complex deep semantic features and features derived from a compositional distributional semantic model. Our own submission to this workshop is fully detailed in Section 3.2. It ranked 10th with a Pearson score of 0.71 on similarity and 8th on entailment, with an accuracy of 78.5% (Gupta et al., 2014).

SemEval 2015 Task 2: Semantic Textual Similarity

In 2015, the STS shared task returned as Task 2: Semantic Textual Similarity. This time the shared task involved 3 subtasks (Agirre et al., 2015):

- Subtask 1: English STS
- Subtask 2: Spanish STS
- Subtask 3: A pilot subtask on interpretable STS. An attempt to add an explanatory layer to the similarity score.

The organisers did not create a new dataset for training, opting instead to use training data from previous workshops. For testing, however, the workshop organisers put together samples from 5 different texts, resulting in the following categories of test data:

(1) Image description (image): This subset is derived from Flickr dataset (Rashtchian et al., 2010). It consists of images from Flickr, depicting actions and events of people or animals, with five captions per image.

(2) News Headlines (headlines): Derived from headlines mined from several news sources by European Media Monitor[5].

(3) Student answers paired with reference answers (answers-students) derived from the BEETLE corpus (Dzikovska et al., 2010),

(4) Answers to questions posted in stack exchange forums (answers-forum), derived by pairing answers to the same questions or different questions.

(5) English discussion forum data exhibiting committed belief (belief), collected from DEFT Committed Belief Annotation dataset

A small subset of the test set, along with its gold standard scores, was also released as trial data.

The English subtask in particular attracted 29 teams who submitted 74 runs overall. The baseline system had a 0.69 mean Pearson score, whereas the best ranked system scored a mean Pearson score of 0.8 across all the different test sets. The highest ranking system was a supervised machine learning system which used word alignments and similarities between sentences as features (Sultan et al., 2015). The second best system had a Pearson correlation of 0.794 and was submitted by Hänig et al. (2015). Similarly to the best system, it used a combination of word alignments and string-based features with a support vector regressor to measure similarity. Our contribution to the shared task achieved a score of 0.721 and is further described in Section 3.3 (Béchara et al., 2015).

A Spanish subtask was also included. This subtask attracted 7 participants with 16 overall runs, 67% of which outperformed the baseline. The final subtask, on

[5] `http://emm.newsexplorer.eu/NewsExplorer/home/en/latest.html`

interpretable STS, asked participants to include an explanatory layer. These tasks were not relevant to this thesis.

SemEval 2016 Task 1: Semantic Textual Similarity

The STS task was part of SemEval 2016's Task 1: Semantic Textual Similarity, Monolingual and Cross-Lingual Evaluation (Agirre et al., 2016). Here the organisers included cross-lingual STS for the first time, using Spanish and English sentence pairs. This task had 2 subtasks:

- Subtask 1: English STS
- Subtask 2: Cross-lingual Subtask (English–Spanish)

Meanwhile, the pilot subtask introduced in 2015 became its own task (Task 2: Interpretable Semantic Textual Similarity).

All datasets released during prior STS evaluations were available as trial and training data. The new test data was collected from a diverse set of sources:

- The newswire headlines collected from Europe Media Monitor (Best et al., 2005)
- Sentences collected from the Corpus of Plagiarised Short Answers (Clough and Stevenson, 2011)
- MT translations of news data and their post-edited counterparts from the EAMT 2011 corpus (Specia, 2011)
- Question-question and answer-answer evaluation sets extracted from the Stack Exchange Data Dump (Stack Exchange, Inc., 2016).

The English subtask attracted 43 participating teams with a total of 119 submitted runs. The best overall performance was by Samsung Poland NLP Team's EN1 system, with an overall Pearson correlation of 0.778 (Rychalska et al., 2016). This system stresses the importance of a diverse set of features and methods to capture semantic similarity. In addition to using similarity features with an SVM classifier, the authors further augment their scores with those produced by a bi-directional Gated Recurrent Neural Network and additional features. The primary limitation of their system, however, is its heavy reliance on word order, which makes its performance situational. That did not stop it from outperforming all the other systems submitted to the shared task. The second best system, UWB, achieved a Pearson correlation score of 0.757 (Brychcín and Svoboda, 2016). UWB built a SVM regression model with a variety of features based on lexical, syntactic, and semantic information. Comparatively, the baseline system, based on a simple vector representation, achieved an overall Pearson score of 0.511. We submitted a system that builds on our previous work and expands it to use word embeddings and dense vector space LSTM based sentence representations (Béchara et al., 2016). Our system outperformed the baseline and ranked 22 out of 43 participating systems with an overall Pearson correlation of 0.694.

This year's workshop also saw the introduction of a cross-lingual subtask, using English–Spanish sentence pairs rather than the usual monolingual sentence pairs of previous tasks. The datasets were generated using previous STS annotations from the datasets from previous years, with one sentence in each sentence pair translated into Spanish by human translators.

SemEval 2017

At the time of writing of this thesis, SemEval 2017 is the last workshop to offer the classical STS shared task. In 2017, Task 1 included cross-lingual and monolin-

gual pairs: (Arabic-English, Spanish-English, Arabic-Arabic, English-English and Spanish-Spanish), and urged its participants to focus on systems that could participate in more than one pairing (Cer et al., 2017). Organisers put together a new dataset for evaluation, combined from the Stanford Natural Language Inference (SNLI) (Bowman et al., 2015) and data from the WMT 2014 quality estimation track (Bojar et al., 2014). The task saw strong participation with 31 teams (84 submissions), of which 17 participated in all tracks. The top performing system was ECNU (Tian et al., 2017), a universal model for STS, that translates non-English sentences into English (using MT) and then determines their semantic similarity. The authors use a combination of traditional NLP methods and deep learning to determine the STS scores. Their traditional NLP methods follow those of previous models in extracting a set of effective features and then using supervised machine learning regressors to predict the STS score. However, they also use neural network methods with distributional representations of sentences in order to obtain separate similarity scores. The final scores (73.16) are obtained by averaging both scores, and outperformed the baseline system (53.7). In second place, with an overall score of 67.89, is BIT (Wu et al., 2017), a system which primarily uses sentence information content (IC) informed by WordNet and BNC word frequencies.

Between 2012 and 2017, SemEval and its shared task on STS has been pioneering research in the field, providing a venue for the evaluation of state-of-the-art methods and algorithms. The workshop has provided STS datasets which researchers can use for testing and comparison. The availability of these datasets enabled other researchers to develop STS systems independently from the evaluation conferences. Recent years have seen a shift to Deep Learning methods (Ramaprabha et al., 2018). The systems submitted to the shared task have been among the most cutting edge in the field, and have provided an essential contribution to the research presented in this thesis.

2.3 Machine Translation Evaluation

As machine translations become more wide-spread, the need to evaluate these systems and the quality of their output becomes more and more important. The evaluation of MT output is a highly complex task, due to the ambiguous nature of natural language and the vast differences in how these languages express concepts (Han and Wong, 2016). Developers rely on a variety of techniques to assess the quality of machine translation output. Human evaluation, while in theory reliable and able to give the best view of the system's performance, is also costly, time-consuming, and inconsistent due to the subjectivity of human judgement. This renders it inefficient in the context of larger corpora. In light of these short-comings, automatic evaluation tools have been developed to estimate the quality, defined in the broad sense, of MT output. Automatic evaluation metrics assess MT translation systems without relying on the costly and often unreliable judgement of human evaluators. Automatic evaluation metrics can be divided into two subgroups: reference-based evaluation metrics, which rely on one or more reference translations (usually provided by human translators) to produce MT output, and reference-free evaluation metrics, which rely solely on the source and hypothesis translation for assessment.

In the rest of this section we cover an overview of evaluation techniques both manual and automatic. We detail the recent developments in both reference-based and reference-free evaluation techniques and assess their correlation with human judgement, their reliability, and their short-comings.

2.3.1 Human Evaluation

Human evaluation is based on the manual scoring of machine translation output, taking into account two aspects of correctness: fluency and adequacy. Fluency measures the readability and understandability, while adequacy concerns itself with

whether or not the translated sentence conveys the original sentence's meaning. Evaluators, or judges, are usually asked to assess each sentence with two separate ranks between 1–5, one for fluency and one for adequacy, usually along a scale such as the one presented in Table 2.2 (Koehn and Monz, 2006). These human-produced measures reflect the usability and appropriateness of MT output. More often than not, these judgements are used to compare systems to each other rather than rate a system on its own. Therefore, it is much more common for judges to rank a number of systems.

Table 2.2: Commonly used evaluation scale for human judges

	Adequacy	Fluency
5	All Meaning	Flawless
4	Most Meaning	Good
3	Much Meaning	Non-native
2	Little Meaning	Disfluent
1	None	Incomprehensible

In Vilar et al. (2006), the authors manually analyse statistical machine translation output in order to identify the system's main errors and present a classification framework of these errors. Using the output of the RWTH Statistical Machine Translation system (Vilar et al., 2005), the authors classify the errors into five main categories:

- **Missing Words**: When a word in the generated translation is missing. This category is further subdivided into essential and non-essential words. Essential words will usually alter the meaning of sentences.
- **Word Order**: When words or phrases appear in the wrong order in the generated translation. This category is divided into short-range and long-range order.

- **Incorrect Words**: Usually the largest of the categories. Incorrect words can be words for which the system chooses the incorrect translation or for which the system failed to disambiguate a word correctly. It also applies to the wrong form or inflection of words, bad stylistic choices or literally translated idiomatic expressions.

- **Unknown Words**: These are words for which the system cannot find a translation at all, or when the system does not recognise characters.

- **Punctuation**: These are minor punctuation errors.

The error categories need not be mutually exclusive, and one error can have more than one category assigned to it. The full error taxonomy, presented in Vilar et al. (2005), has been used by many other researchers to date. While exhaustive, this kind of error topography can be difficult to use consistently. Bojar (2011) use a version of this topography to evaluate English-Czech machine translation, using 18 native speakers of Czech. The authors found inter-annotator agreement of only 39%. They attribute this low agreement to two factors: First, the annotators disagree on what the ideal target should be, and second they disagree on what edits need to be made to reach that ideal target.

Another common method for human assessment utilises post-editing. This is usually done by comparing the raw translation to its final post-edited product. This is even more time-consuming than the ranking system proposed earlier, and is highly dependent on the evaluator's skill and judgement. Such is the case of the human translation error rate metric (HTER) (Snover et al., 2006). HTER measures the minimum number of edits (insertions, deletions, substitutions and shifts) required to make the translation acceptable. The minimum number of edits is itself determined automatically, much like the automatic metrics we will address in Section 2.3.2. In that sense, HTER is a hybrid metric, utilising both automatic methods and human input. As the human input needs to be provided after translation, however, we have

chosen to include HTER under human evaluation.

Crowdsourcing is another type of manual evaluation that attempts to address the cost and time problem that it faces. Crowdsourcing involves using users on the Web (often anonymous) to evaluate or rank sentences the way a professional would. Unlike professionals, however, crowdsourcing is low-cost and quicker. However, unlike with professional translators, the quality of evaluators and their familiarity with the domain is not guaranteed. The anonymity of the Web also leads to problems with spammers. Despite these shortcomings, researchers have shown that agreement rates for crowd-sourced non-experts are comparable to those of professionals (Zaidan and Callison-Burch, 2011). Furthermore, crowd-based system ranking has a very strong correlation with expert-based ranking (Bentivogli et al., 2011; Goto et al., 2014).

However, human evaluation comes with its own set of problems. To begin with, it is much more time-consuming than automatic metrics. Furthermore, its results are subjective and biases among human judges call for a variety of normalisation methods before the numbers are usable. Agreement between annotators is frequently found to be low when it comes to machine translation evaluation (Callison-Burch et al., 2008), with kappa figures reported as low as 0.25 for adequacy. Despite this, when done properly this type of evaluation has the potential to be the most reliable type of evaluation, as it allows a much more fine-grained analysis than most metrics, and opens up the possibility of error analysis. Furthermore, automatic metrics are often found not to reflect translation quality as perceived by humans, as will be made evident in the next two sections.

2.3.2 Reference-Based Evaluation Metrics

Reference-based evaluation metrics compare MT output to a reference translation, which is a translation provided by a human and considered to be a "gold stan-

dard" translation. The assumption is that the score returned would mimic human judgement, as the closer the output is to the human "gold standard", the higher its quality. At times, several translations of the same text are used to account for the fact that the same text can be translated in different ways.

BLEU (Papineni et al., 2002) is a popular and widely used score for MT evaluation that relies on n-gram overlapping to approximate human judgements, and is currently the most used metric to evaluate machine translation systems. BLEU matches n-grams between the MT output and the reference translation, using n-gram precision with a brevity penalty as the score, as demonstrated in Equation 2.3.

$$\text{BLEU}(n) = \prod_{1}^{n} PREC_i{}^{\frac{1}{n}} \cdot bp \tag{2.3}$$

where n is the order of n-gram, $PREC_i$ is the i-gram precision and bp is the brevity penalty. The brevity penalty, defined in Equation 2.4, is added to stop shorter sentences receiving too high a score:

$$bp = exp(max(\frac{len(Ref)}{len(Out)} - 1, 0)) \tag{2.4}$$

where $len(Ref)$ is the length of the reference and $len(Out)$ is the length of the output. To account for language variability, BLEU normally makes use of multiple reference translations.

Criticisms of BLEU and n-gram matching metrics in general are addressed by Callison-Burch et al. (2008), who show that BLEU fails to correlate to (and even contradicts) human judgement. BLEU is very sensitive to small changes in the output, and fails to capture linguistic variations, especially in the case where only one reference translation is being used. Furthermore, metrics such as BLEU are specifically designed for system or corpus-level assessment, and do not fare well when evaluating quality on a sentence-level. Several smoothing techniques have been proposed to make BLEU, work at a sentence level (Lin and Och, 2004). Despite

criticisms, BLEU remains the most widely-used used automated metric, as it is efficient, inexpensive, and easy to use.

Other n-gram lexical metrics include NIST, which uses the arithmetic mean instead. NIST is specifically designed to improve on BLEU, and is based on BLEU. In order to prevent the inflation of SMT evaluation scores, it focuses on common words and high confidence translations, NIST weights n-grams based on their rarity. Unlike BLEU, shorter sentences do not impact NIST scores as dramatically. NIST has been shown to outperform BLEU specifically in the case of Chinese translations (Doddington, 2002).

Further reference-based evaluation metrics rely on edit distance rather than n-gram overlap. Su et al. (1992) introduce Word Error Rate (WER), a metric that determines the editing that a human post-editor would have to perform to change a system output so it matches the given reference translation. This metric is defined by Equation 2.5.

$$\text{WER} = \frac{\#INS + \#DEL + \#MOD}{len(Ref)} \tag{2.5}$$

Where INS, DEL and MOD represent the number of insertions, deletions and substitutions required to make the output identical to the given reference translation. Criticisms of WER point to its inadequacy in taking word order into account. TER (Snover et al., 2006) addresses this weakness, however. TER is another edit distance metric that is defined in Equation 2.6:

$$\text{TER} = \frac{\#INS + \#DEL + \#MOD + \#SHIFT}{len(Ref)} \tag{2.6}$$

where SHIFT represents the number of sequence shifts required.

TER is similar to HTER (c.f. Section2.3.1), but it uses a reference translation

instead of requiring human input to post-edit. Snover et al. (2006) show that a single-reference variant TER performs as well as a four-reference variant of BLEU, and that human-targeted TER correlates better with human judgements than its n-gram based counterparts.

Like n-gram based metrics, edit-distance metrics face serious limitations in that they rely heavily on reference translations, limiting their flexibility. If an automatic translation fails to match a given reference translation, it will be penalised by the metric even if it is a fully fluent and adequate translation. While multiple references mitigate this problem somewhat, it is practically impossible to cover every single possible translation for a given input.

The previously mentioned metrics make relatively few attempts account for semantic information for MT evaluation. METEOR (Banerjee and Lavie, 2005) rectifies this short-coming by matching unigrams based on more than just the overlap between the words presented in the n-grams. Much like the other metrics mentioned here, METEOR is based on unigram matching between the machine translation and reference translation. It computes a score based on a combination of all generalised unigram matches between the two strings have been found, METEOR computes a score for this matching using a combination of unigram-precision, unigram-recall, and a measure of fragmentation that captures how well-ordered the matched words in the machine translation are in relation to the reference. METEOR also uses paraphrases to capture the many ways in which a translation can be expressed. According to the authors, this metric shows improved correlation with human judgements.

In Giménez and Màrquez (2007), the authors propose metrics which take linguistic features at more abstract levels into account. They show that metrics based on deeper linguistic information make up for the short-comings of automatic evaluation metrics and produce more reliable system rankings that better correlate with human judgement. Their metric is based on shallow semantic structures such as word forms,

part of speech tags, dependency relationships, syntactic phrases semantic roles and named entities. They call these structures linguistic elements (LE), and posit that a sentence can be seen as a bag of linguistic elements. Their system outperforms metrics based on lexical matching alone. However, they find that semantic oriented metrics are more stable at system level rather than at sentence level.

Lo and Wu (2011) argue that reference-based metrics such as BLEU do not adequately capture semantic correctness between the machine translation output and the reference translation. They define a good translation as one that preserves the central information, rather than focusing on fluency. They present their alternative, MEANT, a semi-automatic metric that assesses translations by matching semantic role fillers. MEANT, however, is semi-automatic, as it relies on human judgement to determine the correctness of these semantic role-fillers. For this reason, it is more efficient and less labour-intensive than pure manual evaluation.

Castillo and Estrella (2012a) follow in this line of research, claiming that the output of machine translation systems will correlate more strongly with human translations if they have a higher Semantic Textual Similarity score with the reference translation. Using a machine learning approach based on 8 sentence-level semantic features, they determine a semantic similarity score between each output segment and its corresponding reference translation. They report competitive scores at system-level, concluding that their metric is useful for measuring the performance of MT systems.

More recent metrics have been proposed that have adapted deep learning methods. One such metric is ReVal (Gupta et al., 2015a). ReVal uses dependency-tree Long Short Term Memory (LSTM) network to represent both the hypothesis and the reference with a dense vector. ReVal performed competitively, when tested on WMT2013 and WMT2014 data, outperforming BLEU on a system level for 5 out of 6 language pairs.

Machine translation metrics remain an active area of research, with new metrics proposed to compete with the metrics described in this section. The Workshop for Machine Translation includes a shared task for machine translation metrics with several new systems proposed each year (Callison-Burch et al., 2007, 2008, 2009, 2010, 2011, 2012; Bojar et al., 2013, 2014, 2015, 2016, 2017). These workshops usually focus on European languages and have both evaluation and translation tasks.

Despite their wide adoption by the MT community, automatic metrics face a serious limitation as reference translations may not always be available. Providing these references requires skilled translators who can be expensive and time-consuming. Reference translations require previously translated parallel corpora, which means that while they can be used to evaluate the general performance of a system, they cannot be used to evaluate a text that has not been previously translated. While metrics like Bleu and Ter are frequently used to assess the quality of a given system, there are situations where they are not practical for general evaluation of MT output.

2.3.3 Reference-Free Evaluation

The restrictions and short-comings of the reference-based translation metrics have led into further investigation of the MT evaluation problem. Reference-free evaluation was proposed as a solution to the problems introduced by the need for a reference translation. Confidence estimation (CE) in MT treats evaluation as a confidence problem, and measures how sure a system is of its own output. CE is based on techniques used to measure confidence in speech recognition. This score can be interpreted as a quantitative estimate of translation quality (Kulesza and Shieber, 2004) and generally relies on a feature vector which encapsulates information about the hypothesis text, and predicts a variable that indicates the quality of the translation. This variable can range from a binary value that indicates whether the

translation is good or bad, or a continuous score that assesses the translation quality. The features in question depend on the source text and hypothesis translation and do not require a reference translation. This enables testers to use confidence estimation where the reference translation is unavailable at testing time.

Confidence Estimation

While previously used in speech recognition, the use of confidence estimation for machine translation output was first introduced by Gandrabur and Foster (2003), who use CE in order to enhance an interactive text prediction tool for translators. The system provides up to 5 machine translation options from an MT system, based on the first few letters typed by the translator. Using two types of neural nets (single and multi-layer perceptrons), they investigate the benefit of using confidence estimation in discrimination power and the relevance of various features and model combinations. They train their model on the Hansard English-French parallel corpus which spans 1.3 million translation predictions. They found that CE layer provided a significant gain (10% benefit in accuracy) to the translators in two out of three translation models tested.

Ueffing et al. (2003) present and apply several concepts of confidence measures for SMT and compute word posterior probabilities based solely on surface-based features contained in the output. They perform confidence estimation on the word level, measuring the confidence of correctness for each generated word and comparing it to a threshold. If the word's confidence is above the threshold, it is tagged as "correct". Otherwise, it is tagged as "false". They perform their experiments on two separate corpora: a trilingual corpus (LC-STAR) and the TransType2 corpus (Langlais et al., 2000), which consists of technical manuals. The first corpus includes English, Spanish and Catalan, and the second includes English, French, Spanish and German. Using an IBM-4 translation model (Brown et al., 1993), they set up

experiments for nine language pairs and calculated a baseline for correctness based on word error rate (WER), graph error rage (GER) and word graph density (WGD). Experiments showed that both systems performed well and reduced the confidence error rate.

A more detailed study of confidence estimation for machine translation is presented by Blatz et al. (2004), who use MT metrics to evaluate the "correctness" of MT output on both the sentence level and the word level. They label sentences as "good" and "bad"; based on WER, a metric based on Levenstein distance, and NIST scores rather than manual scores, and estimate the quality of the output by analysing a total of 91 sentence-level features in the source and target texts. These features were chosen to account for different aspects of translation, and included both surface features (such as average target word statistics and basic syntactic information) and system-dependent features, such as n-best lists and IBM model scores (which model probability distributions of translations) (Brown et al., 1993). In keeping with the principle of confidence estimation, none of the features relied on the reference translation. They carried out their experiments using Chinese-to-English datasets and tested different machine learning learning algorithms on this dataset. They tested a number of machine learning and found that multilayer perceptrons outperformed all the other models. However, the system performed poorly, due to its reliance on automatic metrics rather than human evaluation.

Quirk (2004) avoided this problem by using human scores to tag their dataset. While the ensuing data set was considerably smaller, the scores were more reliable and therefore more useful than those based on automatic metrics, which often fail to correlate to human judgement. Quirk (2004) showed that a smaller dataset with indicative features that are manually evaluated can outperform large samples with automatic scores.

Confidence estimation (CE) focuses mainly on system-dependent features and

on measuring how confident a given system is rather than how correct the translation is. This approach has several disadvantages. The confidence estimation relies heavily on the machine translation system, and is limited in versatility and application. Furthermore, confidence estimation is useless in cases where the system does not grant access to its inner workings, such as commercially available systems. Additionally, extraction of system-dependent features can be computationally costly, more so than system-independent features.

Machine Translation Quality Estimation

Machine Translation Quality Estimation (MTQE) addresses the short-comings of confidence estimation and investigate the use features and measures that are system-independent. Though nowadays, MTQE has come to refer to both CE and QE prediction systems. This allows MTQE to make use of the information in both sets of features.

Early work in MTQE built on the concept of confidence estimation described in Section 2.3.3. In contrast, MTQE used only system-independent features based on the source sentence and target translation (Specia et al., 2009a). They trained a Support Vector Machine (SVM) regression model based on 74 shallow features, and reported significant gains in accuracy over MT evaluation metrics. At first, these approaches to MTQE focused mainly on shallow features based on the source and target sentences. Such features include n-gram counts, the average length of tokens, punctuation statistics and sentence length among other features. Later systems incorporate linguistic features such as part of speech tags, syntactic information and word alignment information (Specia et al., 2010).

In the context of MTQE, the term "quality" itself is flexible and can change to reflect specific applications, from quality assurance, gisting and estimating post-editing (PE) effort to ranking translations. Specia et al. (2009b) define quality

in terms of PE efficiency, using MTQE to filter out sentences that would require too much time to post-edit. Similarly, He et al. (2010) use MTQE techniques to predict human PE effort and recommend MT outputs to Translation Memory (TM) users based on estimated PE effort. In contrast, Specia et al. (2010) use MTQE to rank translations from different systems and highlight inadequate segments for post-editing.

The WMT Shared Tasks on Quality Estimation

Since 2012, MTQE has been the focus of a shared task at the annual Workshop for Statistical Machine Translation (WMT). This task has provided a common ground for the comparison and evaluation of different MTQE systems and data at the word, sentence and document level. The workshop is one of many similar workshops and focuses on using parallel corpora for machine translation, evaluation, and quality estimation by setting up shared tasks with an open participation, and ranking the participating systems according to performance.

The first task, in 2012, set out two variations: ranking and scoring. The dataset itself was made up of English-Spanish language pairs, produced by the phrase-based SMT system, Moses (Koehn et al., 2003). The sentences were manually annotated for post-editing effort using the following scale:

1 A score of 1 indicates the The MT output is incomprehensible and needs to be translated from scratch.

2 A score of 2 means that about 50% -70% of the MT output needs to be edited.

3 A score of 3 means about 25-50% of the MT output needs to be edited.

4 A score of 4 only about 10-25% of the MT output needs to be edited.

5 A score of 5 means the MT output requires little to no editing.

Systems were required to predict a score based on this scale for English-Spanish translations. Evaluation for the scoring task was performed using Root Mean Squared Error (MSE) and Mean Absolute Error (MAE).

MAE measures the average magnitude of the errors on the test set, without considering their direction. Therefore, it is ideal for measuring the accuracy for continuous variables. MAE is calculated as per Equation 2.7.

$$MAE = \frac{1}{n} \sum |x_i - y| \tag{2.7}$$

where n is the number of instances in the test set, x_i is the score predicted by the system, and y is the observed score. Root Mean Squared Error is similar, but uses the square as defined by the Equation 2.8.

$$RMSE = \sqrt{\frac{1}{n} \sum (x_i - y)^2} \tag{2.8}$$

To evaluate the ranking task, the organisers developed a new metric called DeltaAvg (Callison-Burch et al., 2012). The baseline system to beat was a system using the 17 features found most relevant in Specia et al. (2009b). These features would later become the baseline for QuEst (c.f. Section 2.3.4). Eleven teams submitted one or more systems to the shared task. Only 5 of the submitted systems beat the baseline by a statistically significant measure. The top performing system was submitted by the SDL Language Weaver team (Soricut et al., 2012), and used both a M5P regression-tree and SVM-regression models. In addition to the baseline features and the system-dependent (decoder) features, the team developed and tested a variety of features including out of vocabulary words, Language Model Perplexity scores and word alignment scores.

The shared task on quality estimation returned in 2013 (Bojar et al., 2013).

The [1-5] scores were replaced with HTER scores (c.f. Section 2.3.1). The task also added two additional subtasks, a system selection subtask, where participants were required to rank up to five alternative translations for the same source sentence produced by multiple MT systems, and a subtaks predicting post-editing time, where the participants were required to predict a time (in seconds) required to post-edit the MT output. Additionally, a whole new word-level quality estimation task was introduced. In their submission to the workshop, Biçici and Van Genabith (2013) introduce a novel approach in the form of referential translation machines, a computational model for identifying the translation acts between any two data sets. RTMs remove the need to use any system or language dependent data, and perform competitively in both the sentence and word level MTQE tasks. This system ranked first and second in all of the subtasks.

The following year, the WMT shared task on MTQE aimed to study the effects of the new labels and focus more on the effects of mixed domains datasets, as well as the effects of MTQE on human translations. That year's shared task had three subtasks: The first was a MTQE task based on a new scoring system of post-editing effort (a scale of 1-3, where 1 means use as is, 2 means post-editing is required, and 3 means translate from scratch). The second and third subtasks were similar to those of the previous year: MTQE based on HTER and MTQE based on post-editing time respectively. The language pairs were expanded to include English-Spanish, English-German, and Spanish-English. Most notably, that year's task focused on system-independent features only, and did not provide the participants access to system-dependent information as in previous years (Bojar et al., 2014). That year's top submissions vastly outperformed the baseline system, showing how far the research has come since the first WMT shared task in 2012.

The WMT shared task on quality estimation continued in the following year, with the aim to MTQE further with larger datasets (Bojar et al., 2015). This time the three subtasks were divided along the sentence, word and the new document level

MTQE task. A phrase-level MTQE task was added in WMT 2016 (Bojar et al., 2016) and 2017 (Bojar et al., 2017). While most of the work on MTQE presented in this chapter so far is feature based, designed for the output of SMT systems, more recent neural solutions to MTQE have been proposed. POSTECH, the top performing system to the WMT2017 shared task on MTQE was entirely neural based, and required no feature engineering at all (Kim et al., 2017). The system uses Multi-level task learning with stack propagation. This system extended to the word, phrase, and sentence-level MTQE tasks and outperformed all other systems in these tasks. The rise of neural MT and the application of neural methods to MTQE led to the development of deepQuest, a framework similar to QuEst, created to accommodate neural approaches at all levels, including the document level (Ive et al., 2018). Their system, while based on a simplification of POSTECH, outperforms the former while also being significantly faster.

2.3.4 The QuEst Framework

Today, the state of the art quality estimation techniques have been combined into the open-source framework, QuEst++ (Specia et al., 2015). QuEst++ is an open source framework for machine translation quality estimation. In addition to a feature extraction framework, QuEst++ provides the scikit-learn toolkit [6], which contains all the machine learning algorithms necessary to build the prediction models.

The tool offers three different variants for MTQE:

- Sentence-Level MTQE: compares a pair of sentences in the source of target language. This variant has received the most attention in research to date.
- Document-Level MTQE: predicts a single label for entire documents.
- Word-Level MTQE: produces a label for each target word

[6] `https://scikit-learn.org/stable/`

QuEst++ gives access to a large variety of features, each relevant to different tasks and definitions of quality. The features are divided into three categories:

- Baseline Features: These are the 17 system-independent features. These features include sentence length, n-gram overlap, and punctuation tokens.
- Black Box Features: An extended list of system-independent features. There 111 Black Box features in total. These include the percentage of nouns and verbs in each sentence, language model probabilities and perplexities and the percentage of numbers per sentence, among others.
- Glass Box Features: These are system-dependent features specific to the SMT system. There are 47 in total. These include distortion features, the percentage of incorrectly translated words and log probability scores, among others.

QuEst++ and its predecessor, QuEst (Specia et al., 2013), are used as a baseline in the earlier WMT shared tasks for Quality Estimation (QE) shared tasks. In this research, we use QuEst++ for all MTQE tasks.

2.4 Motivation and Context

Given the success of supervised machine learning methods in tackling the problem of similarity, we chose to build on this trend when designing our own model for determining semantic textual similarity. Choosing to work with supervised machine learning allowed us to focus on feature engineering to identify the strongest indicators of similarity. Furthermore, at the time of writing, the lack of a large enough dataset to for similarity tasks made it difficult to use deep learning methods. In this section, we will provide a brief overview of the design choices and justify their use.

2.4.1 Support Vector Machines

We chose to use Support Vector Machines (SVM) to address our machine learning solutions and continue to use SVMs throughout our research. SVM is a supervised machine learning algorithm that is capable of both classification and regression (Vapnik, 2013). In classification problems, SVMs classify sets of data by determining an optimal hyperplace that separates the data into categories. This can be done for highly complex data and can be extended to non-linear data via the Kernel Trick. Support vectors were considered state-of-the-art in solving classification and regression problems (Lee et al., 2010) because of its good generalisation performance in many real applications. At the time of writing, SVMs were widely used in NLP tasks and QuEst++ (c.f. Section 2.3.4) was run using SVMs. In order to compare our results directly to those produced by the WMT workshops and the Quest++ baseline system in particular, we chose to continue to use to SVMs throughout this research. This choice allowed us to finely tune our features based on performance, and to achieve competitive results despite small amounts of data at our disposal. In order to achieve good results with SVMs, the hyperparameters need to be tuned. For a RBF kernel, these parameters are namely C and gamma. C is the "cost" of misclassification, and trades off correct classification against maximising the decision function's margin. Gamma defines how far the influence of a single training example reaches. A high value of gamma leads to more accuracy but potentially biased results. For every SVM model, these parameters need to be tuned for optimal variance and bias. In order to determine the optimal values, we perform a Gridsearch.

All the training, prediction and tuning were run using LibSVM[7] (Chang and Lin, 2011a), a freely available and open-source integrated software for support vector classification and regression.

[7] `https://www.csie.ntu.edu.tw/~cjlin/libsvm/`

2.4.2 Evaluation Methods

Most of our systems were submitted in shared tasks, in order to compare our work to the upcoming state of the art methods in each of the respective fields. The evaluation methods these tasks use can differ based on the different workshops and organisers. In order to remain comparable, we use the same methods used by the shared tasks.

Pearson Correlation Coefficient

As a general rule we use Pearson Correlation Coefficient to evaluate our methods. Pearson Correlation Coefficient measures the linear correlation between two variables, as presented in Equation 2.1. Pearson is one of the most widely used coefficients to measure linear relationships between two normal distributed variables, and is used in most of the shared tasks to rank submissions.

We use the Pearson Correlation Coefficient to evaluate our STS method and compare it to the other systems submitted to the SemEval tasks in 2014 and 2015 shared tasks. We also use Pearson to perform feature selection.

Spearman Rank Correlation

While we do not use Spearman Rank Correlation for evaluation, it is used once in this thesis, when calculating distributional similarity measures in Section 3.3.1.

Spearman Correlation differs somewhat from Pearson as it sorts the observations by rank and computes the distance between rank rather than absolute. Spearman is considered more robust to outliers and is not linked to the distribution of the data. Spearman Correlation is defined by Equation 2.9

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)} \tag{2.9}$$

where d is the pairwise distance of the ranks of the variables and n is the number of samples.

Mean Error

The Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) are metrics used to measure accuracy. MAE measures the average magnitude of errors in a set of predictions as described in Equation 2.7. RMSE is similar but uses the roof of the squared differences as described in Equation RMSE. Both methods express average model prediction error and are used by shared tasks such as WMT (c.f. Section 2.3.3) and QATS ((Štajner et al., 2016)) to evaluate submitted systems and rank them. Therefore, we use MAE to evaluate our approach to integrating STS into the machine translation quality estimation in Section 4.3 and both MAE and RMSE to evaluate our approach to using STS for text simplification in Section 6.2.

2.5 Conclusion

In this chapter, we presented related research in STS and MT Evaluation. In order to properly frame context of STS research, we started by presenting the evolution of RTE and its applications. The majority of research into STS is presented within the context of SemEval 2012 – 2017, where shared tasks on STS provided a venue for the comparison and evaluation of STS systems. STS also provides a large variety of annotated data within a number of different domains for testing training. We present the shared tasks on STS and the top-performing systems in each workshop. With a

few exceptions, most of these systems frame STS as a supervised machine learning task, with a variety of language technology and semantically motivated features. Newer submissions have ventured into deep learning to determine STS. The second half of this chapter focuses on MT evaluation, covering both human and automatic metrics, like Bleu, Ter and Meteor. Automatic metrics work well as long as one or more reference translation is available against which to compare the translation. MTQE does not require a reference translation, and instead relies on information present in the source and translated sentences, and occasionally on the MT system itself. MTQE treats evaluation as a machine learning task, building regressors or classifiers depending on the task, and relying on a variety of both system dependent and independent features. We present the evolution of MTQE from confidence estimation, mostly within the context of WMT 2011 – 2018. Finally, we describe QuEst++, a widely-used feature extractor for sentence-level, document-level and word-level MTQE.

Chapter 3

Determining Semantic Textual Similarity through Machine Learning

3.1 Introduction

In Section 2.2, we defined STS as the degree of semantic similarity between two sentences. As this thesis investigates the uses of STS in evaluation, we dedicate this chapter to developing a ML system that captures semantic similarity between two sentences. As we intend to use this system in several experiments, we focus on building a system that is accurate, system-independent, and fast in terms of run time. This chapter describes three systems that evaluate semantic textual similarity through machine learning methods and are part of our submissions to the SemEval Workshop for 2014 (Marelli et al., 2014a) and 2015 (Agirre et al., 2015). The remainder of this chapter is structured as follows: Section 3.2 describes our participation in the first task of SemEval 2014 workshop, titled: *Evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual*

entailment. Section 3.3 presents our participation in 2015's Task 2: *Semantic Textual Similarity*. The systems build on each other and attempt to improve on the previous system by introducing new features that capture semantic similarity more accurately. Section 3.4 describes the system we use for experiments in the remainder of this thesis: a simplification of the previous two systems optimised for speed and accuracy.

3.2 UoW Submission, SemEval2014

This system is our submission to the SemEval 2014 Task 1, which required participants to submit systems that predicted the semantic similarity between two sentences. This task is further detailed in Section 2.2.

We submitted system runs for both sub-tasks, using the same overall system with minor variations for each. Both systems employ a Machine Learning (ML) method which exploits available NLP technology, typed dependencies, paraphrasing, machine translation evaluation metrics, quality estimation metrics and corpus pattern analysis[8] (CPA). However, while we build a regression model for relatedness, we treat the entailment problem as a classification model. Both systems use the same set of features. According to our evaluation, some features perform better depending on the specific subtask.

The rest of this section describes in detail the training data, features, and Machine Learning algorithms before reporting the performance results and error analysis.

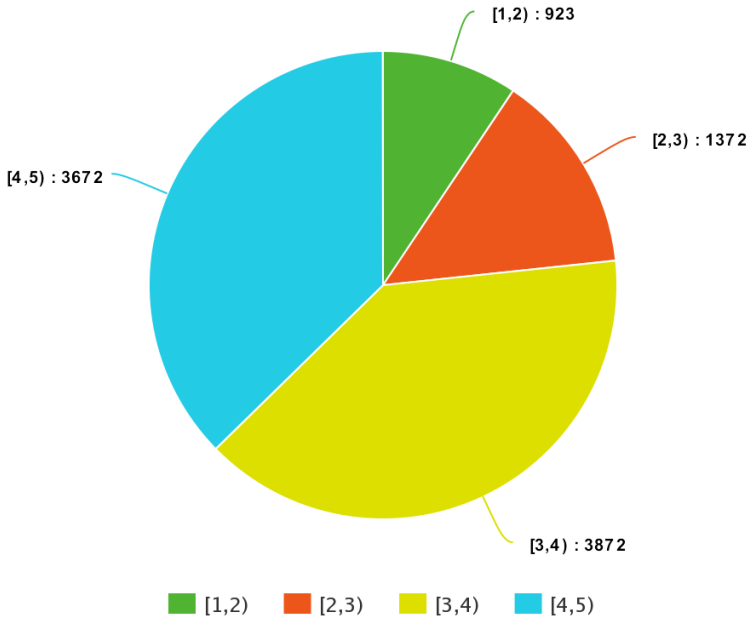
[8] `http://pdev.org.uk`

3.2.1 The SICK Dataset

The SICK (Sentences Involving Compositional Knowledge) dataset (Marelli et al., 2014b) is a set of aligned sentences specifically designed for compositional distributional semantics. It includes a large number of English sentence pairs that are rich in lexical, syntactic and semantic phenomena. The similarity score is a score between 1 and 5, previously described in Section 2.2.3. These scores are obtained by averaging several separate annotations by distinct evaluators. For this reason, they are continuous, rather than discrete.

The SICK dataset is generated from existing datasets based on images and video descriptions. Each sentence pair is annotated for relatedness (similarity) and entailment by means of crowd-sourcing techniques. It consists of 10,000 pairs. In the full set, the gold scores' distribution for relatedness are summarised in Figure 3.1.

Figure 3.1: Distribution of Gold Scores for Relatedness in SICK



We provide examples for each of these intervals below in Examples 1-5.

In Example (1), the two sentences are unrelated, except in that they both concern

children.

(1) Range: [1,2) – Score 1.6

a. Sentence A: There are no children playing and waiting

b. Sentence B: Three Asian kids are dancing and a man is looking

In Example (2), the two sentences are on the same topic, but the details are different.

(2) Range: [2,3) – Score 2.4

a. Sentence A: A man is sitting on a chair and rubbing his eyes

b. Sentence B: A tattooed man is on a sofa and is holding a pencil

In Example (3), the two sentences are on the same topic and only some minor details (the clothes and the sound equipment), differ.

(3) Range: [3,4) – Score 3.2

a. Sentence A: A girl is wearing white clothes and is dancing

b. Sentence B: The blond girl is dancing in front of the sound equipment

In Example (4), the two sentences are virtually identical, differing only slightly.

(4) Range: [4,5] – Score 4.8

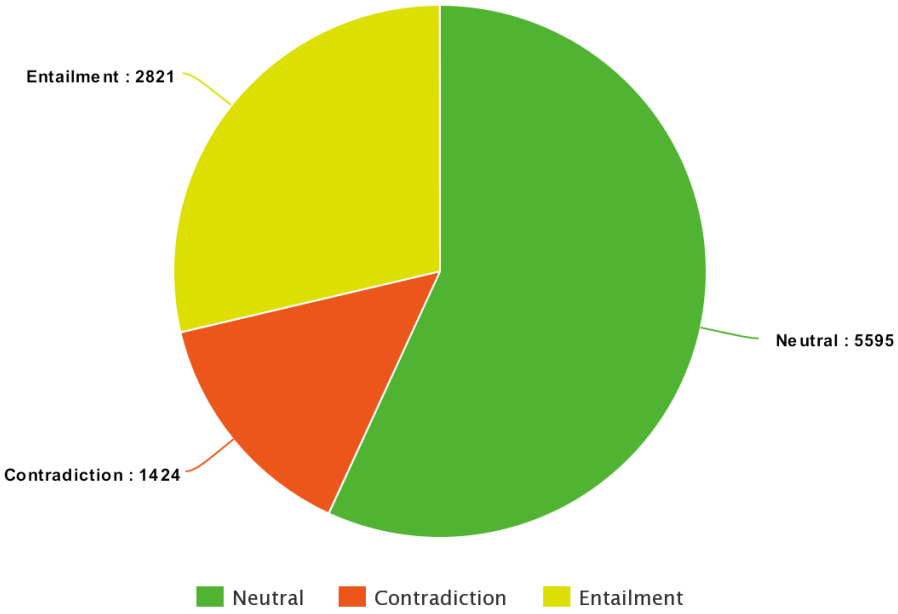
a. Sentence A: A motorcyclist is riding a motorbike dangerously along a roadway

b. Sentence B: A motorcyclist is riding a motorbike along a roadway

In the full set, gold scores' distribution for entailment are summarised in figure 3.2.

We provide examples for the entailment distribution below in Examples 5-7:

Figure 3.2: Distribution of Gold Scores for Entailment in SICK



(5) Class: Neutral

a. Sentence A: Two dogs are playing by a tree

b. Sentence B: A dog is leaping high in the air and another is watching

(6) Class: Contradiction

a. Sentence A: A hiker is on top of the mountain and is dancing

b. Sentence B: There is no hiker dancing on top of the mountain

(7) Class: Entailment

a. Sentence A: A nude lady is walking in front of a crowd in body paint

b. Sentence B: A topless girl is covered in paint

In addition to the STS task, the SICK corpus is employed in further experiments through-out this thesis.

3.2.2 The Feature Set

This system uses a total of 31 features which are expanded on in this subsection. The features are divided into 6 subsets based on their type. The first 7 features are language technology features extracted using off-the-shelf language processing tools.

The rest of this section goes into further detail on the features and the methods through which they were extracted.

Language Technology Features (F1 - F7)

The language technology features refer to a set of 7 figures that calculate word overlap between two sentences. The aim of these features is to capture token-based grammatical similarity between a pair of sentences. We extract a total of 7 language technology features. These features use pre-existing language processing tools which are found in the Stanford CoreNLP[9] toolkit (Manning et al., 2014). In a general sense, these features calculate word overlap between two sentences using different types of units that constitute sentences.

The features look at more than just the surface form of sentences. Some features look at the overlap of parts of speech, lemma, dependency relations and named entities. These features are useful in that they encapsulate a surface form similarity between sentences, and capture which words, concepts, and actions recur in a pair.

Overlap is computed using the Jaccard similarity coefficient. The Jaccard similarity coefficient is defined as the measure of similarity between two sets. It is calculated by taking the size of the intersection of two sets divided by the size of their union.

This is demonstrated in equation 3.1.

[9] `http://nlp.stanford.edu/software/corenlp.shtml`

$$Sim(s1, s2) = \frac{|s1 \cap s2|}{|s1 \cup s2|} \tag{3.1}$$

where $Sim(s1, s2)$ is the Jaccard similarity between sets of words $s1$ and $s2$.

The 7 language technology features can be summarised as:

1. **Word Overlap**

   We calculate 4 different types of word overlap:

   (a) Surface form overlap (F1): determines word overlap based on the unchanged form of the word. This feature captures how many words overlap between sentences.

   (b) Lemma form overlap (F2): converts words to their base forms before calculating overlap. By looking at the base forms of words, this feature can capture similarity even when words have different inflections, affixes, suffixes, or tenses across sentences.

   (c) Part of Speech overlap (F3): annotates words with their part of speech before calculating overlap.

   (d) Named Entities Overlap (NE) (F4): the Stanford Toolkit identifies named entities, which we then use to calculate overlap.

2. **Dependency Relations (F4-F5: GOVDep)**

   The first of two features based on dependency relations, this feature concatenates words involved in a dependency relation.

   For example, the sentence: *the kids are playing outdoors* becomes *kids::the, playing::kids, playing::are, ROOT::playing, playing::outdoors*

   The Jaccard similarity is then calculated using the concatenated dependent words.

3. **Grammatical Relations (F6: GRAMrel)**

   The second of the two dependency relations uses morphosyntactic tags instead of concatenated dependency words. In this case the same sentence used above (*the kids are playing outdoors*) is replaced by its grammatical tags and becomes *det, nsubj, aux, root, dobj" and "det, nsubj, aux, root, dobj.* Jaccard Similarity is then calculated using these tags instead of the original words.

4. **Coreference (F7:CORef)**

   The final language technology feature identifies coreference relations and determines clusters of coreference entities. This feature attempts to capture where expressions and phrases refer to the same entity. This can help us derive the correct interpretation of texts across a sentence pair.

   The coreference feature value is calculated using equation 3.2:

$$Coref = \frac{CC}{TC} \tag{3.2}$$

   where:

   CC is the number of clusters formed by the participation of entities (at least one entity from each sentence of the pair) in both sentences and TC is the total number of clusters.

Paraphrasing Feature (F8: PPD)

The paraphrasing feature aims to detect when a segment is a paraphrase of another segment. The assumption behind this feature is that if segments of the text are paraphrases of each other, then we would expect a high similarity between the sentence pairs. To that end, it makes use of the PPDB paraphrase database (Ganitkevitch et al., 2013) to extend each sentence's n-grams with matching n-grams from the database. We then calculate overlap (Jaccard similarity) between these n-

grams to get a feature value.

Machine Translation Evaluation Features (F9–F11)

In another attempt to capture similarity between two sentences, we turn to BLEU (Papineni et al., 2002). BLEU matches n-grams between the MT output and the reference translation, using n-gram precision. We use SBLEU to capture the overlap on the sentence level. BLEU and SBLEU are further explained in Chapter 2.3.2. We extract 3 features using BLEU based on the sentences' surface form (SBLEU), lemma (LBLEU) and parts of speech (PBLEU). Note the similarity between the forms here and the forms in the Language Technology features. These features provide another layer of similarity based on n-grams overlap rather than word overlap.

Corpus Pattern Analysis Features (F12: CPA)

Corpus Pattern Analysis (CPA) is a corpus-driven technique in corpus linguistics and lexicography that associates word meaning with word use by mapping meaning onto specific syntagmatic patterns exhibited by a verb in any type of text (Hanks, 2013). CPA aims at identifying patterns of normal usage ('norms'), including literal and metaphorical uses, phrasal verbs and idioms, and exploring the way patterns are creatively exploited ('exploitations'). CPA is currently being used to compile the Pattern Dictionary of English Verbs (PDEV), an online lexical resource that currently covers nearly 1,300 English verbs. Our final two features make use of the Pattern Dictionary of English Verbs. The first of these features returns 1 when the verb patterns across sentences match, and 0 otherwise. The second feature returns a probability of a PDEV pattern given a specific word. The probability itself is computed over a manually tagged portion of the British National Corpus (BNC).

Negation Feature (F13: NEG)

This feature checks for the presence of a negation word (which we define to be: *no, never* and *not*) in the pair of sentences and returns "1" ("0" otherwise) if both or none of the sentences contain any of these words. This feature helps determine if one sentence is a direct contradiction of another and therefore proves more important in modelling textual entailment and contradiction than semantic textual similarity.

Machine Translation Quality Estimation Features (F14–F31: QE)

We include seventeen features based on Machine Translation Quality Estimation (MTQE) features used by Specia et al. (2009b) to predict machine translation quality without the use of a reference translation. All of these features are extracted using QuEst, an earlier version of QuEst++ (c.f. Chapter 4.4.1). To make these features work for our data, which is monolingual, we treat the first set of sentences as the Machine Translation (MT) "source", and the second set of sentences as the MT "target". The MTQE features include shallow surface features such as the number of punctuation marks, the average length of words, the number of tokens, n-gram frequencies and language model probabilities[10]. MTQE features relate to well-formedness and syntax, and are not usually used to compute semantic relatedness between sentences. However, they do reflect structural similarities between sentences and therefore might have given us a small insight into semantic relatedness.

3.2.3 Prediction and Results

We submitted runs to both subtasks in SemEval2014's Task 1, tackling both the relatedness and entailment problem. Therefore, we built two separate supervised machine learning models using the same features, using Support Vector Machines for classification and regression analysis. We rely on Lib-SVM (Chang and Lin, 2011b) to build our models.

[10]A full description of the framework is provided in Section 4.4.1

We build a regression model for the relatedness task and a classification model for the entailment task. The regression model estimates a continuous score between 1 and 5 for each sentence. As the entailment task had only 3 discrete categories, we use a 3-way classification model for it instead. We trained both systems on the 4,500 sentence training set, augmented with the 500 sentence trial data provided by the SemEval workshop task organisers. To cope with the overfitting of the data, we optimised the cost parametres through a grid-search which uses a 5-fold cross-validation method.

Our system performed adequately, with our best run achieving a mean Pearson Correlation of 0.72 and a MSE of 0.51. In comparison, the highest ranking system scored a Pearson Correlation of 0.82 and MSE of 0.32, while the lowest scored a Pearson Correlation of 0.47 and a MSE of 1.10. Our system performed similarly in the entailment task, achieving an accuracy of 78.53% on our best run, compared to the baseline of 56.2%. The best performing system achieved an accuracy of 84.5%. Tables 3.1 and 3.2 provide a summary of these results. These tables include all 4 runs we submitted to the tasks, compared to the baseline (basic word overlap) and the best performing system.

Table 3.1: Semantic Textual Similarity - as calculated by SemEval2014

	Pearson Correlation					
	Run 1	**Run 2**	**Run 3**	**Run 4**	**Best System**	**Baseline**
C	8	8	2	2		
λ	0.0441	0.0441	0.125	0.125		
Pearson	0.7111	0.71	0.70	0.70	0.82	0.63

Table 3.2: Entailment - as calculated by SemEval2014

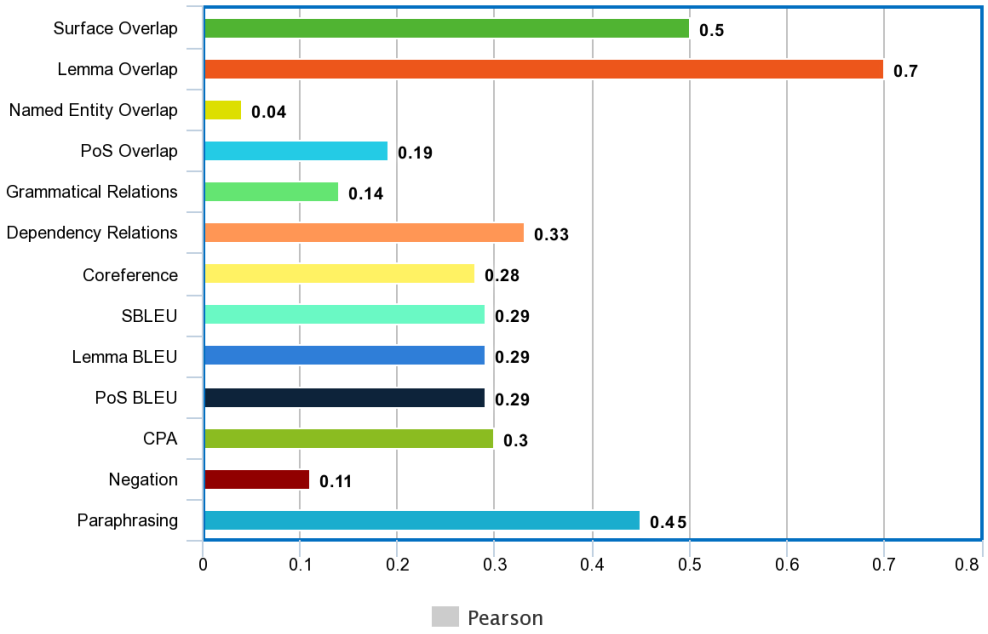
	Accuracy					
	Run 1	**Run 2**	**Run 3**	**Run 4**	**Best System**	**Baseline**
C	16	16	8	8		
λ	0.0625	0.0625	0.5	0.5		
Accuracy	78.53	78.53	78.34	78.34	84.5	56.2

3.2.4 A Feature Analysis

We find the least useful features to be the MTQE features, as they focus on fluency rather than semantic correctness. The MTQE features contribute to only minor improvements, increasing the Pearson correlation coefficient by only 0.027.

The most useful features, in contrast, are language technology features, which prove to be the strongest predictors. The lemmatised word overlap is the strongest indicator for similarity, followed by the surface form word overlap ("surface" on the chart). The paraphrasing feature also shows a good correlation to similarity, with a Pearson score of 0.45. The CPA features and the machine translation evaluation features (SBLEU, LBLEU, PBLEU) show a weak correlation when tested individually. The negation feature (NE) does not seem to correlate with relatedness. Figure 3.3 summarises the performance of all the features individually, showing the best performing features for the similarity task.

Figure 3.3: Individual Performance of features



3.3 MiniExperts, SemEval2015

The following year, we participated in SemEval2015's Task 2, titled "Semantic Textual Similarity". Similar to Task 1 in the previous year, this task called for systems that, given a sentence pair A and B, return a similarity score based on how closely related the two sentences are. However, where 2014's Task 1 included an entailment subtask, this year's task instead was divided into an English and Spanish language subtask.

For our submission to the 2015 SemEval workshop, we improve on our 2014 system. We strip the UoW (c.f. Section 3.2) system down to a baseline system with 13 features, choosing the features that performed the most strongly based on the feature selection algorithm. These features are language technology features, the paraphrasing features and the CPA features, or features 1 through 13 in the previous system.

3.3.1 The Feature Set

The features in this submission build on the features for the 2014 submission. The basis of this system are the first 13 features described in Section 3.2: The language technology features, the paraphrasing feature, the CPA features, the negation feature and the machine translation evaluation features. In addition to the 13 baseline features, we introduce a set of Distributional, Semantic and Conceptual Similarity Measures, as well as a feature reflecting MWEs across sentences.

Distributional Similarity Measures

We use two independent IR measures, the Spearman's Rank Correlation Coefficient (SCC) and the χ^2 to compute the similarity between two sentences written in the same language (Kilgarriff, 2001). For every pair of sentences, we use the lemmas to extract the list of common terms to compute both measures.

Conceptual Similarity Measures

In order to calculate the conceptual similarity, we take advantage of the BabelNet[11] (Navigli and Ponzetto, 2012) multilingual semantic network, which organises lexical information conceptually. To that end, we create a conceptual sentence for all input pairs by extracting lemmatised nouns, verbs, adjectives and adverbs. We then build a conceptual term list of all the occurrences of the term in the conceptual network (i.e. BabelNet). The resulting "conceptual representation" of both sentences, contains a set of conceptual term lists. For each term in the "conceptual_sentence_1", we count the number of co-occurrences in the conceptual term lists in the "conceptual_sentence_2". After computing all the co-occurrences, we used these values to calculate the Jaccard' (Jaccard, 1901), Lin' (Lin, 1998) and PMI' (Turney, 2001) scores.

[11] `http://babelnet.org`

This feature is especially expensive computationally, as it required us to query the BabelNet database for every sentence.

Semantic Similarity Measures

This feature takes advantage of the Align, Disambiguate and Walk (ADW)[12] library (Pilehvar et al., 2013), a WordNet-based approach for measuring semantic similarity of arbitrary pairs of lexical items. As the ADW library permits us to measure the semantic similarity between two raw English sentences, either by using disambiguation or not, we used both options to calculate all the comparison methods made available by the library, i.e. WeightedOverlap, Cosine, Jaccard, KLDivergence and JensenShannon divergence. These values make up 5 different features.

Multiword Expressions

Multiword Expressions (MWEs) are meaningful lexical units whose distinct idiosyncratic properties call for special treatment within a computational system. For the purpose of our experiments, we focused on two more common types of MWEs in English: `verb noun` combinations (e.g. *make sure, take place*) and `verb particle` constructions (e.g. *to make up*, *to put down*).

Whenever a `verb+noun` or a `verb+particle` combination occurs in our sentence pair, we search a prepared list MWEs, sorted according to their likelihood measures of association. The degree of association of these combinations served as a feature in our ML system.

[12] `http://lcl.uniroma1.it/adw`

3.3.2 Prediction and Results

We build a regression model which estimates a continuous score between 0 and 5 for each sentence pair.

We trained this system on a combination of training and trial data provided by the 2012, 2013 and 2014 SemEval tasks. We used these datasets to form a training set of 9,750 sentence pairs combining the different domains covered by the STS task: image description (image), news headlines (headlines), student answers paired with reference answers (answers-students), answers to questions posted in stack exchange forums (answers-forum) and English discussion forum data exhibiting committed belief (belief).

We used LibSVM[13], a library for SVMs developed by Chang and Lin (2011b) in order to predict the semantic similarity. We optimised for the values of C and γ through a grid-search which uses a 5-fold cross-validation method, and all systems use an RBF kernel. Our system performed adequately, with our best run achieving a mean Pearson Correlation of 0.7216, as scored by SemEval 2015. Table 3.3 provides a breakdown of these results. In comparison, the top ranking system achieved a mean Pearson Correlation of 0.8015.

Table 3.3: Pearson Correlation - as calculated by SemEval2015

	Pearson Correlation
answers-forums	0.6781
answers-students	0.7304
belief	0.6294
headlines	0.6912
images	0.8109
mean	***0.7216***
rank (out of 74)	***33***

[13] `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`

For the sake of comparison, we also tested our system on the same dataset used in the 2014 workshop. These features bring a small improvement to the table, and our new system tested on the SICK dataset yielded a better Pearson correlation score. As reported in Table 3.4, the MiniExperts system submitted to the 2015 workshop drastically outperformed the UoW system. However, the new features, especially the distributional and conceptual similarity features, are computationally far too costly to realistically run on large data sets, making them useless in spite of the improvements.

Table 3.4: Comparing Results - UoW vs MiniExperts on the SICK dataset

	Pearson Correlation	
	UoW	**MiniExperts**
Pearson	0.7166	0.806

3.4 The STS system used in this research

As this thesis concerns itself with STS and its uses in various applications, we decided to use an optimised version combining both STS systems for the rest of the experiments described here-in. We therefore performed some feature selection in order to determine which features we could safely discard without dramatically affecting the system's accuracy.

Our final features are selected for both accuracy and efficiency. As the 17 Quality Estimation Features and the Multiword Expression features scored the lowest, with a Pearson Correlation less that 0.1, we discard them from the new system. We also discard the distributional and Conceptual Similarity measures for being too computationally costly and time-consuming.

The final system consists of 13 features:

- Language Technology Features
  - (1-4) Word Overlap: Surface form, Lemma, PoS, and Named Entities.
  - (5) Dependency Relations (GOVDep)
  - (6) Grammatical Relations (GRAMrel)
  - (7) Coreference (CORef)
- (8) Paraphrasing Feature (PPD)
- Machine Translation Evaluation Features
  - (9) Sentence Level BLEU (SBLEU)
  - (10) Lemma Level BLEU (LBLEU)
  - (11) PoS BLEU (PBLEU)
- Corpus Pattern Analysis Features (CPA)
  - (12) CPA Match
  - (13) CPA Probability

These final 13 features described in this section form the basis for the system used to calculate STS throughout the rest of this thesis.

3.5 Conclusion

In this chapter, we presented two fairly efficient and accurate approaches to predicting semantic similarity for English. We build on research mainly submitted to the SemEval workshop's shared tasks on detecting semantic relatedness and entailment.

In Section 3.2, we described our submission to the SemEval 2014 workshop Task 1. In this submission, we use a total of 31 features and an SVM regression model

to assign a continuous score (1-5) to English sentence pairs. This score measures semantic relatedness. One noticeable point of our approach is that we used the same features to also measure entailment, with a 3-way classification model (ENTAILMENT, CONTRADICTION, NEUTRAL). The system performed averagely in the SemEval2014 STS task and achieved a Pearson correlation of 0.711, while the highest ranking system in the workshop achieved a score of 0.828. Our system ranked 10 out of 17 teams that participated in the task. Though we used the same features for both tasks, our system performed well in each of these tasks. Therefore, our system captures reasonably good models to compute semantic relatedness and textual entailment.

In Section 3.3, we presented our submission to the SemEval 2015 workshop, while building on the work presented in Section 3.2. For this submission, we reused the first 13 features from the 2014 submission and added a number of features derived from distributional and conceptual similarity. This version performed reasonably well and the system's best result ranked 33 among 74 submitted runs with 0.722 Pearson correlation over five test sets (only 0.08 correlation points less than the best submitted run).

Finally, we described a streamlined version of the STS systems that optimises for both accuracy and speed. This system uses only 13 features common to both the 2014 (UoW) system and the 2015 (MiniExperts) system, and is the STS system used in the research described in the remainder of this thesis.

Chapter 4

Semantic Textual Similarity in Machine Translation Quality Estimation

4.1 Introduction

The previous chapter presented several Machine Learning approaches for STS on the basis of which we proposed the STS method used in this thesis. In this chapter, we detail a series of experiments in which we use semantic similarity measures to correlate semantic similarity with machine translation quality. We attempt to answer the research question:

RQ1: Can semantic textual similarity help accurately predict the quality of MT output?

We expland the question to ask: Is it possible to evaluate the quality of a translated sentence for which we do not have a reference translation, based on the level of

semantic relatedness between itself and a different sentence for which we do have a reference translation?

Machine Translation Quality Estimation (MTQE) predicts the quality of machine translation output without the need for a reference translation. This quality can be defined differently based on the task at hand. In an attempt to focus further on the adequacy and informativeness of translations, we integrate features of semantic similarity into QuEst (Specia et al., 2015), a framework for MTQE feature extraction. By using the methods we described in Section 3.4, we use semantically similar sentences and their quality scores as features to estimate the quality of machine translated sentences. We propose a set of features that compares MT output to a semantically similar sentence, that has already been assessed, using monolingual STS tools to measure the semantic proximity of the sentence in relation to the second sentence. Our experiments show that finding semantically similar sentences for some datasets is difficult and time-consuming. Therefore, we opt to start from the assumption that we already have access to semantically similar sentences. Our results show that this method can improve the prediction of machine translation quality for semantically similar sentences.

This chapter is structured as follows: the next section, Section 4.2, grants some background information about previous attempts to incorporate semantic information into machine translation quality estimation, in order to frame our research within the context of quality estimation. Section 4.3 describes our method of integrating STS into the evaluation pipeline by using independently evaluated semantically similar sentences. Section 4.4 details the data and tools we use in the experiments described in this chapter. Section 4.5 details three separate sets of experiments on three different datasets and their results. Finally, Section 4.6 sums up our findings.

4.2 Background

Section 2.3.3 presented an overview of MTQE within the context of reference-free evaluation of machine translation output. As we have previously discussed, there have been a few attempts to integrate semantic similarity into MT evaluation (Lo and Wu (2011); Castillo and Estrella (2012b)). The results reported are generally positive, showing that semantic information is not only useful, but often necessary, in order to assess the quality of machine translation output. Different authors have taken different approaches as to how to bring semantics into this task.

Specia et al. (2011) focused on meaning preservation and tried to address the quality of MT output based on how much it preserves the meaning of the source, rather than how easy it is to post-edit. The authors identified a number of system-independent features that focus on adequacy. These features included comparing sentence lengths, word and phrase overlap, named entities, dependency relations, depth of syntactic trees, among others. The authors then trained a multiclass classifier using an SVM model to predict an adequacy score between 1 and 4, depending on the level of meaning preserved in the translation. The dataset was obtained by automatically translating a number of Arabic newswire texts, using two state-of-the-art phrase-based SMT systems. The resulting English sentences were then annotated for adequacy by 2 Arabic-English professional translators. Their obtained results yielded an improvement over a majority class baseline.

The concept of adequacy has also been used by Rubino et al. (2013), who used topic models for MTQE. The authors expanded on the work by Specia et al. (2011) by adding topic model features that focused on content words in sentences. Using the same dataset developed by the authors in Specia et al. (2011) above, the authors reported results outperforming state-of-the-art approaches where the dataset was annotated with adequacy information.

Also in 2013, Biçici (2013) introduced the use of a computational model for judging monolingual and bilingual similarity: the Referential Translation Machines (RTMs). According to Biçici and Yuret (2011), "RTMs provide a computational model for quality and semantic similarity judgments using retrieval of relevant training data". The authors participated in all 4 challenges of the Quality Estimation task at WMT 2013 (Bojar et al., 2013), which included both sentence and word-level MTQE for German to English and English to Spanish translation pairs. The authors used the language models provided by the task organisers to build RTMs for the different language pairs, and trained machine learning models using both ridge regression and support vector regression. RTMs achieved state-of-the-art results and the authors reported top performance in both sentence and word level tasks of the WMT 2013.

In their submission to the WMT2014 shared task on Quality Estimation, de Souza et al. (2014) proposed a set of features that used word alignment information with the aim of addressing semantic relations between sentences. The authors used these features to augment QuEst's 79 black box features and participated in both the word-level and sentence-level subtasks, achieving top results in both. In the same year, Kaljahi et al. (2014) employed syntactic and semantic information in MTQE achieving an improvement over the baseline when combining these features with the surface features of the baseline. The papers presented in this section informed our work, which focuses on the necessity of semantic information for MT adequacy.

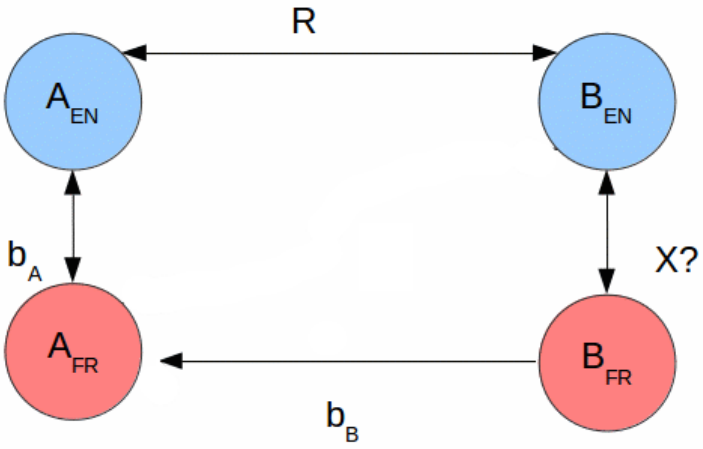
4.3 Integrating Semantic Textual Similarity into Machine Translation Quality Estimation

Previous work has suggested that integrating semantic information into the quality estimation pipeline is a crucial step towards improving the detection of adequacy in

translated sentences. To that end, we propose bridging STS and MTQE by adding semantic textual similarity features into the quality estimation task. As our STS tools rely on monolingual data, we employ the use of a second sentence that bears some semantic resemblance to the sentence we wish to evaluate.

Our approach is illustrated in Figure 4.1, where sentences A_{EN} and B_{EN} are two semantically similar sentences with a similarity score R. Our task is to assess the quality of sentence B_{FR} with the help of sentence A_{FR}, which has already undergone machine translation evaluation, either through post-editing (e.g measuring post-editing effort) or human evaluation (e.g. assessed on a scale from 1–5). As both sentences, A_{EN} and B_{EN} are semantically similar, our hypothesis is that their translations are also semantically similar and thus we can use the reference of sentence B_{FR} to estimate the quality of sentence A_{FR}.

Figure 4.1: Predicting the Quality of MT Output using a Semantically Similar Sentence B



For each sentence A_{EN}, for which we wish to estimate MT quality, we retrieve a semantically similar sentence B_{EN} which has been machine translated and has a reference translation or a quality assessment value. We then extract the following three scores (that we use as STS features):

Semantic Textual Similarity (STS) score:

R represents the STS score between the source sentence pairs (sentence A_{EN} and sentence B_{EN}). This is a continuous score ranging from 0 to 5. We calculate this score using the system described in Section 3.4 in all but one of our experiments, where we already have human annotations about STS.

Quality Score for Sentence A_{FR}:

We calculate the quality of the MT output of Sentence A. This is either a S-BLEU (c.f. Section 2.3.2) score based on a reference translation, or a manual score provided by a human evaluator. This score is labelled b_A in Figure 4.1.

S-Bleu Score for Sentence A:

We have no human evaluation or reference translation for Sentence B_{FR}, but we can calculate a quality score using Sentence A as a reference. We use sentence-level BLEU (S-BLEU) (c.f. Section 2.3.2). This score is labelled b_B in Figure 4.1. S-BLEU is designed to work at the sentence level and will still positively score segments that do not have a high order n-gram match.

4.4 Data and Tools

We use the following open source tools and freely available corpora to design and test our experiments. However, as these tools do not always fulfil our purpose, we develop a number of tools and datasets of our own design as well.

4.4.1 The QuEst Framework

In Section 2.3.4, we provided an overview of the QuEst framework and its various iterations. In our experiments, we use the 17 black-box features from Quest as a baseline to allow for comparison of our work to a well-used and tested baseline. The baseline features are system independent and include shallow surface features (e.g. number of punctuation marks, average length of words, number of words, etc.). They also include n-gram frequencies and language model probabilities. Table 4.1 enumerates QuEst's 17 baseline features.

4.4.2 Translation Model

All our experiments require machine translated output to test our evaluation systems. To that end, we use a phrase based statistical machine translation (PBSMT) system called Moses (Koehn et al., 2007). We build 5-gram language models with Kneser-Ney smoothing trained with SRILM (Stolcke, 2002), a toolkit for building language models, the GIZA++ implementation of IBM word alignment model 4 (Och and Ney, 2003), with refinement and phrase-extraction heuristics as described in (Koehn et al., 2003). We used minimum error rate training (MERT) (Och, 2003) for tuning on the development set.

We trained on 500,000 randomly sampled sentences from the Europarl corpus (Koehn, 2005), and then tuned (using MERT) on 1,000 different unique sentences. We trained two separate models, one to translate from English to French, and one to translate from French to English.

Table 4.1: Quest Baseline Features

	Feature Description
1	Number of tokens in the source sentence
2	Number of tokens in the target sentence
3	Average source token length
4	Language Model probability of the source sentence
5	Language Model probability of the target sentence
6	Average number of occurrences of translated word within the target sentence
7	Average number of translations per source word in the sentence (as given by IBM 1 table thresholded so that prob(t|s) > 0.2)
8	Average number of translations per source word in the sentence weighted by the inverse frequency of each word in the source corpus
9	Percentage of unigrams in quartile 1 of frequency (lower frequency words) in a corpus of the source language
10	Percentage of unigrams in quartile 4 of frequency (higher frequency words) in a corpus of the source language
11	Percentage of bigrams in quartile 1 of frequency of source words in a corpus of the source language
12	Percentage of bigrams in quartile 4 of frequency of source words in a corpus of the source language
13	Percentage of trigrams in quartile 1 of frequency of source words in a corpus of the source language
14	Percentage of trigrams in quartile 4 of frequency of source words in a corpus of the source language
15	Percentage of unigrams in the source sentence seen in a corpus
16	Number of punctuation marks in source sentence
17	Number of punctuation marks in target sentence

4.4.3 The DGT Translation Memory

The Directorate-General for Translation (DGT) translation memory is a corpus of aligned sentences in 22 different languages created from the European Union's legislative documents (Steinberger et al., 2013). The corpus consists of sentences and their professionally produced translations in 231 language pairs and are produced by highly qualified human translators in specialised domains. Most of the sentences (72%) are originally written in English and then translated. The DGT-TM contains about 38 million translation units, over 2,000 of which are identical across all 22 languages. For the purpose of our research, we focus on the English-French language pair.

4.4.4 The SICK Dataset

SICK (Sentences Involving Compositional Knowledge) is a dataset specifically designed for compositional distributional semantics. It includes a large number of English sentence pairs that are rich in lexical, syntactic and semantic phenomena. Further information on the SICK dataset is presented in Section 3.2.1.

4.4.5 The FLICKR EN-FR Dataset

The datasets above face some limitations as they are not specifically designed with our purpose in mind. Either these datasets do not share many semantically similar sentences (DGT-TM), or they lack a reference translation or another reliable quality rating (SICK).

The limitations of the datasets above led us to design a new dataset. This dataset consists of a pair of English sentences of variable level of medium to high semantic similarity, and their French machine translations. These sentences are

extracted from the FLICKR Images dataset used for SemEval 2015 STS tasks (c.f. Section 3.3). Each pair has a similarity rating between 4-5, crowd-sourced by human annotators. Furthermore, we provide for each sentence a French translation created by SMT (using a Moses phrase-based model described in Section 4.4.2), with variable levels of translation quality. Our main objective is to build a dataset where the translations (Fr_1 and Fr_2), are assigned a quality rating and a semantic similarity score.

For this purpose we require two types of human annotations:

1 A quality score for each translation, between 1 and 5.

2 A similarity rating for the French sentences produced by the machine translation,

MT Quality Score

We obtain this score through manual evaluation by a professional translator who was asked to score the sentences on fluency and adequacy and give them a rating between 1 (the translation is unusable) and 5 (the translation is very good).

Example (1) represents a sample entry to be annotated.

(1) a. Sentence A
    (i) EN: A group of kids is playing in a yard and an old man is standing in the background
    (ii) FR: Un groupe d'enfants joue dans une cour et un vieil homme est debout dans l'arrière-plan
  b. Sentence B
    (i) EN: A group of boys in a yard is playing and a man is standing in the background

(ii) FR: Un groupe de garçons dans une cour joue et un homme est debout dans l'arrière-plan

STS Score for the MT Sentences

We obtained the STS scores for the translated MT sentences through crowd-sourcing. Several questionnaires were posted online and circulated among French speakers, mostly students. The questionnaires asked students to look at two machine translated sentences and rank them for similarity.

The resulting dataset is made of 1,000 sentence pairs and their 1,000 machine translations, along with 3 separate annotations for each 4 sentences: a quality score for the French MT output, a STS score for the English sentence pair, and a STS score for the French sentence pair.

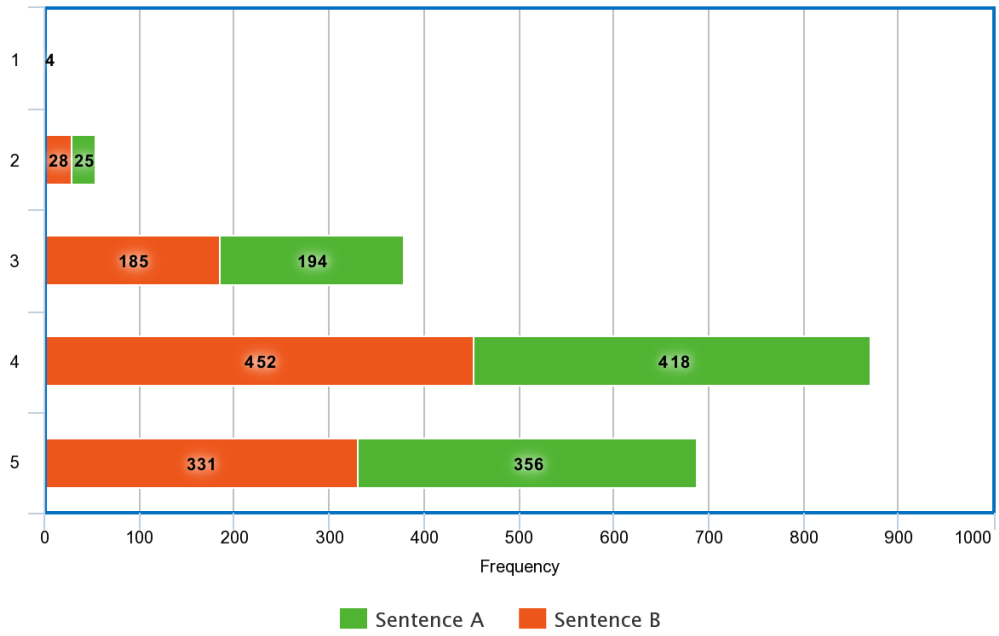
Example (2) shows a sample sentence pair from the dataset with the annotations.

(2) a. Sentence A
    (i) EN: Two children are lying in the snow and are drawing angels
    (ii) FR: Deux enfants sont couchés dans la neige et attirent anges
    (iii) Translation Score: 4
  b. Sentence B
    (i) EN: Two people in snowsuits are lying in the snow and making snow angels
    (ii) FR: Deux personnes dans des habits de neige sont couchés dans la neige et faire des anges de neige
    (iii) Translation Score: 4
    (iv) STS Score: 4

The chart in Figure 4.2 shows a final breakdown of the 1,000 pairs of translations

and their MT evaluation score. As the sentences are quite simple and the MT system performed quite well, most of the sentences (over 75%) score above 4 and less than 25% score 3 or lower.

Figure 4.2: Dataset Statistics



4.5 Experimental Setup

4.5.1 Preliminary Experiments

All our datasets focus on English→French MT output. In all our experiments, we have a set of machine translated sentences A for which we need a quality score and a set of sentences B, semantically similar to the set of sentences A and for which we have some type of evaluation score available.

In early experiments, we attempted to use freely available datasets used in previous workshops on machine translation (WMT2012 and WMT2013) for the translation task and within the news domain (Bojar et al. (2013)). The WMT datasets

have two main advantages: first, they allow us to compare our system with previous systems for evaluation and render our experiments replicable. Second, they have manual evaluations that are available with the machine translations. Each sentence in the WMT dataset comes with a score between 1 and 5, provided by human annotators.

The first obstacle we faced in testing our approach with these datasets was the collection of similar sentences against which to compare and evaluate. We automatically searched large parallel corpora for sentences that yielded high similarity scores. These corpora included the Europarl corpus (Koehn (2005)), the Acquis Communautaire (Steinberger et al., 2006) and previous WMT data (from 2012 and 2013).

Furthermore, the STS system we use (see Section 3.4) returned many false-positives. Some sentences which appeared similar to the STS system were actually too different to be usable. This led to noisy data and unusable results. The scarcity of semantically similar sentences and the computational cost of finding these sentences, led us to look into alternate datasets, preferably those with semantic similarity built into the corpus: the DGT-TM and the SICK dataset.

All our experiments followed similar set-ups. In all cases, we used 500 randomly selected sentences for testing, and the remaining sentences in the respective data-set for training QuEst. We automatically searched large parallel corpora for sentences that yield high similarity scores using the STS system described in Section 3.4.

We attempted to predict the quality scores of the individual sentences, using the STS features described in Section 4.3, added to QuEst's 17 baseline features. We compared our results to both the QuEst baseline (cf. Section 4.4.1) and the majority class baseline[14]. We also tested our STS-related features separately, without the baseline features, and compared them to the system with the combined system

[14]The Mean Absolute Error calculated using the mean rating in the training set as a projected score for every sentence in the test set.

(STS+baseline).

We used the Mean Absolute Error (MAE) to evaluate the prediction rate of our systems. In our experiments, we use S-BLEU scores as the observed score.

4.5.2 Experiments on the DGT-TM

We randomly extract 500 unique sentences (B), then search the rest of the TM for the 5 most semantically similar sentences (A) for each of these 500 sentences (STS score > 3). This results in 2,500 sentences A (500x5) and their semantically similar sentence pairs B. We make sure to avoid any overlap in sentence A so that while semantically similar sentence B might recur, sentence A will remain unique. We assign an STS score to the resulting dataset using the system described in Section 3.4. We then translate these sentence pairs using the translation model described in Section 4.4.2 and use S-BLEU to assign evaluation scores for the MT outputs of Sentence A and B.

Of these 2,500 sentence pairs and their MT outputs, we use 2,000 sentence pairs to train an SVM regression model on Quest's baseline features using using sentence A and its MT output as the source and target sentence. We further use sentence B's S-BLEU score and its STS score with sentence A. We use the remaining 500 sentences to test our system. Example (3) shows a sample sentence (Sentence B) from the DGT-TM along its semantically similar retrieved match (Sentence A) and the machine translation output for each sentence. The STS system gave the original English sentence pair a STS score of 4.46, indicating that only minor details differ.

(3) DG-TM Sample Sentence

a. Sentence A

(i) Source: In order to ensure that the measures provided for in this Regulation are effective , it should enter into force *on the day of*

its publication

(ii) MT: afin de garantir que les mesures prévues dans ce règlement sont efficaces , il devrait entrer en vigueur *sur le jour de sa publication* ,

b. Sentence B

(i) Source: In order to ensure that the measures provided for in this Regulation are effective ,this Regulation should enter into force *immediately*

(ii) MT: afin de garantir que les mesures prévues dans ce règlement sont efficaces , ce règlement doit entrer en vigueur *immédiatement* ,

(iii) STS Score: 4.46

Results:

Our results are summarised in Table 4.2, which shows that the MAE for the combined features (QuEst + STS features) is considerably lower than that of QuEst on its own. A lower MAE indicates a lower error rate, and therefore higher accuracy. This means that the additional use of STS features can improve QuEst's predictive power. Even the 3 STS features on their own outperformed QuEst's baseline features. These results show that our method can prove useful in a context where semantically similar sentences are accessible.

Table 4.2: Predicting the S-Bleu scores for DGT-TM - Mean Absolute Error

System	**MAE**
QuEst Baseline (17 Features)	0.120
STS (3 Features)	0.108
Combined (20 Features)	0.090

4.5.3 Experiments on the SICK Dataset

In order to further test the suitability of our approach for semantically similar sentences, we use the SICK dataset for further experiments. The SICK dataset is generated from existing datasets based on images and video descriptions, and each sentence pair is annotated for relatedness (similarity) and entailment by means of crowd-sourcing techniques (Marelli et al. (2014b)). This means that we did not need to use the STS tool to annotate the sentences.

We extract 5,000 sentence pairs to use in our experiments and translate them into French using the MT system described in Section 4.4.2. The resulting dataset consists of 5,000 semantically similar sentence pairs and their French machine translations. Of this set, 4,500 are used to train an SVM regression model in the same manner as described in Section 4.5.2. The remaining 500 sentences are used for testing.

As the SICK dataset is monolingual and therefore lacking in a reference translation, we opted to use a back-translation (into English) as a reference instead of a French translation for these results. A back-translation is a translation of a translated text back into the original language. Back-translations are usually used to compare translations with the original text for quality and accuracy, and can help to evaluate similarity of meaning between the source and target texts. In machine translation contexts, they can be used to create a pseudo-source that can be compared against the original source. He et al. (2010) used this back-translation as a feature in MTQE with some success. They compared the back-translation to the original source using fuzzy match scoring and used the result to estimate the quality of the translation. The intuition here is that the closer the back translation is to the original source, the better the translation is in the first place. Following this idea, we use the S-Bleu scores of the back-translations as stand-ins for the MT quality scores. We use the MT system described in Section 4.4.2 for the back-translations.

Example (4) shows a sample sentence from the resulting dataset, including the original English sentence pairs and each sentence's MT output. The crowd-sourced STS score for this sentence pair is 4, indicating that only minor details differ.

(4) SICK Sample Sentence

a. Sentence A

(i) Source: Several children are *lying* down and are raising their knees

(ii) MT: Plusieurs enfants sont couchés et élèvent leurs genoux

(iii) Backtranslation: Many children are in bed and raise their knees

b. Sentence B

(i) Source: Several children are *sitting* down and have their knees raised force

(ii) MT: Plusieurs enfants sont assis et ont soulevé leurs genoux

(iii) Backtranslation: Several children sit and have raised their knees

c. STS Score: 4

Results:

Results on the SICK datasets are summarised in Table 4.3. The lowest error rate (MAE) is observed for the system that combined our STS-based features with the QuEst baseline features (Combined (20 Features)) just as in the DGT-TM experiments. We observe that even the STS features on their own outperformed QuEst in this environment.

Table 4.3: Predicting the S-BLEU scores for SICK - Mean Absolute Error

System	**MAE**
QuEst Baseline (17 Features)	0.200
STS (3 Features)	0.189
Combined (20 Features)	0.177

The cherry-picked examples in Table 4.4 are from the SICK dataset, and show that a high STS score between the source sentences can contribute to a high prediction accuracy. In both examples, the predicted score for Sentence A is close to the actual observed score.

Table 4.4: SICK Sample Prediction

	Sentence A	**Sentence B**
Source	Dirt bikers are riding on a trail	Two people are riding motorbikes
MT	Dirt Bikers roulent sur une piste	Deux personnes font du vélo motos
S-BLEU:	0.55 (Predicted) 0.6 (Actual)	0.84
STS	3.6	
Source	A man is leaning against a pole and is surrounded	A man is leaning against a pole and is surrounded by people
MT	Un homme est appuyée contre un poteau et est entouré par des gens	Un homme est appuyée contre un poteau et est entouré
S-BLEU:	0.91 (Predicted) 1 (Actual)	0.91
STS	4.2	

Furthermore, when we filtered the test set for the SICK experiments for sentences with high similarity (4+), we observed an even higher drop in MAE, as demonstrated

in Table 4.5. This suggests that our experiments perform especially well if we select for sentences with high similarity, as we would expect.

Table 4.5: Predicting the S-Bleu scores for SICK sentences with high similarity - Mean Absolute Error

System	**MAE**
QuEst Baseline (17 Features)	0.20
Combined (20 Features)	0.15

4.5.4 The FLICKR EN-FR Dataset

While BLEU is widely used to evaluate MT systems today, it still has severe short-comings when it comes to correlating with human judgement. This severely hinders the performance in the experiments, which rely heavily on sentence-level BLEU scores to evaluate our system.

In order to address these short-comings, we run experiments mirroring those run on the DGT-TM corpus on the new dataset that we designed (c.f. Section 4.4.5). We use 200 randomly chosen sentences as a test set, and use the remaining 800 sentences to train our machine learning system.

As the quality scores for the dataset we designed are discrete (1-5) as opposed to the continuous S-BLEU scores used in previous experiments, we therefore use a SVM classifier rather than a regressor, hoping to more accurately predict our results. Our system is trained to classify sentences into one of 5 different categories, from 1 – 5. Our results show that the addition of the STS-related features can improve our predictions marginally over those of the QuEst baseline features. Table 4.6 shows a 5% increase for Baseline+STS features over the Baseline alone.

Table 4.6: Classification Accuracy for New Dataset

	QuEst Baseline	Baseline + STS
Accuracy	40%	45%

4.6 Conclusion

In this chapter, we sought to answer the first of our research questions (RQ1): Is it possible to evaluate the quality of a translated sentence for which we do not have a reference translation, based on the level of semantic relatedness between itself and a different sentence for which we do have a reference translation? We devised and presented our method to apply semantic textual similarity to machine translation evaluation, using semantically similar sentences and their quality scores as features to estimate the quality of machine translated sentences.

To that end, we introduced 3 semantically motivated features that use a previously evaluated semantically similar sentence, and use these features to augment QuEst's baseline features. After some preliminary experiments, we tested our approach on three different datasets: The DGT-TM, the SICK dataset, and a dataset of our own design based on FLICKR images. The latter we put together by translating semantically similar sentence pairs and evaluating the MT output. The results show improvements over the baseline for all 3 datasets, especially when selecting for high STS sentences. The results are encouraging, showing that these features can improve over the baseline when a sufficiently similar sentence against which to compare is available. We conclude that this approach can be quite useful in settings where we wish to predict the quality of sentences within a very specific domain.

Chapter 5

Quality Estimation in the Translation Workflow: A User Study

5.1 Introduction

In previous chapters we investigated the impact of STS on MTQE and its ability to improve the prediction of MT quality. In this chapter, we investigate the real-world applications of this method and attempt to answer RQ2.

RQ2 To what extent does the use of quality estimation tools affect the efficiency of the translation workflow?

To address this issue, we designed a user study that attempts to successfully integrate MT Quality Estimation into real-life translation workflows. We employed a traffic light system to present translators with different categories of sentences and determine how effective MTQE is at improving the efficiency of the translation workflow. We engaged 4 different professional translators in this task, and measured

the translators' productivity in different scenarios: translating from scratch, post-editing without using MTQE, and post-editing using MTQE. Our results show that MTQE information, when accurate, improves post-editing efficiency.

The remainder of this chapter is structured as follows: the next section, Section 5.2, presents some background information that motivates our design decisions throughout this chapter. Section 5.3 describes the preparation and construction of the dataset used in the user study. Section 5.4 describes the user study and provides the details of the traffic lights system. It also presents an analysis and a discussion of the results obtained. Finally, Section 5.6 sums up our findings in this chapter.

5.2 Background

In recent years, Machine Translation Post-Editing (MTPE) has become more widely used in the translation industry (Zaretskaya et al., 2015; Schneider et al., 2018). In light of this developement, assessing the quality of the MT becomes a more pressing concern. Poor quality MT might end up being more trouble than it is worth, costing a post-editor more time as they assess and rewrite a non-viable suggestion. Many professional translators have acknowledged that consistent low-quality MT can be frustrating, and can lead them to give up on post-editing entirely. Therefore, we posit that a system that assesses the quality of the MT suggestion before it presents it to the translator can help cut down on the time and effort involved for the translator. Machine Translation Quality Estimation (MTQE) can provide this assessment and help the post-editor by proposing for post-editing only sentences which are good enough.

In Section 2.3.3, we presented the concept of reference-free translation and specifically MTQE. QE in MT aims to predict the quality of the MT output without using a reference translation (Blatz et al., 2004; Specia et al., 2011). This field has re-

ceived extensive interest from the research community in recent years, resulting in the proposal of a number of machine learning methods that estimate the quality of a translation on well defined data sets, but which do not necessarily reflect the reality of professional translators. In order to integrate MTQE successfully in translation workflow, it is necessary to know when a segment is useful for a translator. However, and as pointed out by Turchi et al. (2015), "QE research has not been followed by conclusive results that demonstrate whether the use of quality labels can actually lead to noticeable productivity gains in the CAT framework".

Some attempts to investigate the impact of MTQE in post-editing. In a study similar to ours, Turchi et al. (2015) explored whether the use of quality labels can actually lead to noticeable gains in productivity. The authors presented translators with binary quality labels (green to post-edit and red to translate). They used HTER, which we introduced in Section 2.3.1, to determine the labels. The authors denoted a HTER of 0.4 as the boundary between post-editing and translating from scratch, prompting post-editors to discard suggestions with a score under 0.4, and to post-edit suggestions with a score over 0.4 The authors used a modified Mate-Cat (Federico et al., 2014), adapted to provide a single MT suggestion, and a red or green label. They used sentences from an English IT user manual, which they translated into Italian using a PBSMT system, Moses (Koehn et al., 2007). They then post-edited these translations and generated the HTER scores. Their dataset consisted of 1,389 segments, of which 542 were used to train the MTQE engine, and were used for testing. In total, they gathered two instances of each segment, one for the scenario in which the translator was shown the MTQE label, and one in which the translator was not shown the MTQE label. While they observed a slight increase in productivity of 1.5 seconds per word, they concluded that this increase is not statistically significant across the dataset. However, further investigation of their data showed a statistically significant percentage of gains for medium-length suggestions with HTER>0.1.

Moorkens et al. (2015) investigated the extent to which human estimates of post-editing effort predict actual post-editing effort. They also researched how much the display of MTQE scores influenced post-editing behaviour. The authors used two different groups of participants: The first group consisted of six member of staff, postdoctoral researchers and PhD students of a Brazilian University. The second group consisted of 33 undergraduate and Masters translation students. The first step of the study involved asking the first group of participants to assess the quality of a set of 80 segments of two Wikipedia articles. These articles described Paraguay and Bolivia and were Machine Translated into Portuguese using Microsoft's Bing Translator. They were asked to classify the MT output according to a 3-grade scale:

1. Segments requiring a complete retranslation;
2. Segments requiring some post-editing but for which PE is still quicker than retranslation; and
3. Segments requiring little or no post-editing.

The second step of the study took place after a break of at least 2 weeks, in order to allow the participants time to forget their original ratings. The same participants were asked to post-edit the same segments, without showing any type of MTQE information. In the final step of the study, the authors asked the second group of participants (undergraduate and masters students) to post-edit the same sample. This time, however, they included the MTQE information gathered the first step. Although their study is based in a rather small sample, their findings suggest that "the presentation of post-editing effort indicators in the user interface appears not to impact on actual post-editing effort".

In the following year, Moorkens and Way (2016) published a second study compared the use of translation memory (TM) with that of MT among translators. They engaged 7 translators and asked them to rate 60 segments translated from

English into German. The text was taken from two sources with similar domains: the documentation of the an open-source computer-aided design program called FreeCAD, and from the Wikipedia entry describing what computer-aided design is. The authors found that low- and mid-ranking fuzzy matches that are presented to translators without scores, only serve to irritate the translators. Furthermore, the translators found over 36% of such instances useless for their purposes. In contrast, MT matches were always found to be have some utility. Moorkens and Way (2016) concluded that their findings suggest that "MT confidence measures need to be developed as a matter of urgency, which can be used by post-editors to wrest control over what MT outputs they wish to see, and perhaps more importantly still, which ones should be withheld".

In their more recent study, Moorkens and O'Brien (2017) attempt to determine the specific user interface needs for post-editors of MT through a survey of translators. The authors report that 81% expressed the need for confidence scores for each target text segment from the MT engine. This finding validates the impact of our study, as we specifically aimed at investigating the impact of showing MTQE to translators when undergoing MTQE tasks.

Teixeira and O'Brien (2017) investigated the impact of MTQE on the post-editing effort of 20 English to Spanish translators post-editing 4 texts from the WMT13 news dataset. They used four types of scenarios: *No MTQE*, *Accurate MTQE*, *Inaccurate MTQE*, and *Human Quality Estimation*. In contrast to our work, the Quality Estimation data was gathered using the direct assessment method proposed by Graham et al. (2015). Their goal was to determine the impact of the different modes of MTQE on the time spent (temporal effort), the number of keystrokes (physical effort) and the gaze behaviour (cognitive effort). Their results showed no significant differences in terms of cognitive effort. In the case of the average number of keystrokes used or time spent across the different modes of MTQE, there were no significant differences per type of MTQE. However, there were significant differ-

ences when the MTQE score level was higher (the higher the score level, the less time was spent and fewer keys were typed regardless of the MTQE type). They concluded that displaying MTQE scores was not necessarily better than displaying no scores. As we outline later in this chapter, these results contradict our findings, which strongly suggest that MTQE information has a positive effect on reducing the time and effort involved in post-editing.

Our user study followed in the footsteps of these studies, with two major differences. We used FMS instead of more traditional MT evaluation metrics as translators are more used to work with TM leveraging and fuzzy matches (Parra Escartín and Arcedillo, 2015b,a). More significantly, however, our study attempted to identify the difference between the effects of good and accurate MTQE, versus that of mediocre or even inaccurate MTQE.

5.3 Machine Translation Quality Estimation

In Section 2.3.3, we introduced the concept of MTQE and how it estimates the quality of machine translation output without the need for a reference translation. In Chapter 4 we explained our method for integrating STS into MTQE and reported the results based on 3 sets of experiments. In those experiments, quality can be defined differently based on the task at hand. In this chapter, we consider quality related to fuzzy match score (FMS) in line with the findings of Parra Escartín and Arcedillo (2015b,a) who identify a correlation between the editing effort and FMS. For this reason, our MTQE system will predict the FMS between the automatic translation of a sentence, for which the quality needs to be assessed, and its correct translation, without having access to this correct translation.

5.3.1 Autodesk data

In April 2015, Autodesk announced the release of the Autodesk Post-Editing Data corpus. It consists of parallel data where English is the source language and there are up to 12 target languages (simplified and Traditional Chinese, Czech, French, German, Hungarian, Italian, Japanese, Korean, Polish, Brazilian Portuguese, Russian and Spanish). The size per language pair varies from 30,000 segments to 410,000 and each segment is labeled with information as to whether it comes from a TM match or it is MT output. The post-edited target sentences are also included in the dataset. The data belongs to the technical domain, and the segments come predominantly from software manuals.

This corpus was released with MTQE tasks in mind. The fact that it is publicly available makes it a very good choice for our study, as it includes FMS which can be used to train our MTQE. In addition it is domain-specific making it appropriate for our user study where we try to replicate the conditions in which professional translators normally work.

For training and testing our MTQE system, we randomly selected 5,500 sentences: 5,000 sentences were used to train our system, and the remaining 500 sentences were kept for testing.

5.3.2 Evaluation of MTQE

We trained three different systems: In the first system, we used the 79 features extracted using QuEst++ (c.f. Section 4.4.1) with its default language resources and without any additional semantic information provided by our STS method (c.f. Section 4.3). For the second system, we tuned QuEst++ to our dataset by replacing some resources with Autodesk-specific data. We replaced both the English and Spanish corpora with 67,030 sentences from the Autodesk Translation Memory data.

We also built a new Language Model and vocabulary file using this aligned corpus. This tuned QuEst++ to our specific domain and, as the results show, improved the performance of the baseline features.

The third system was the domain tuned version of QuEst++ (second system) enhanced with our STS features. In order to calculate these features, we searched the remaining unused sentences in the Autodesk dataset for sentences with a high similarity to the 5,500 randomly chosen dataset. The results are summarized in Table 5.1.

Table 5.1: MAE predicting the FMS for Autodesk

System Description	MAE
QuEst++ – out of the box	9.82
QuEst++ – tuned for in-domain data	9.78
QuEst++ – with STS features	9.52

The results summarized in Table 5.1 show that QuEst++ performs at better when tuned for the specific dataset and augmented with STS features. However, this tells us little about how useful these predictions would be in an industrial setting. In order to understand the impact of these predictions, we replaced the FMS with a binary label that would tell the translator whether they should post-edit or translate from scratch. We choose a FMS of 75 or higher to be the threshold for post-editing, as per the findings of Parra Escartín and Arcedillo (2015a). The resulting predictions matched up with the observed labels 85% of the time. In a real-world setting, we posit that this accuracy would be of great use to translators and would speed up the process translation process. However, the extent to which this improves the efficiency of the post-editing process would need to be observed in a controlled environment with actual translators

We took a closer look at the cherry-picked examples from the Autodesk test

set. Examples (1), (2) and (3) show the prediction matching quite closely with the observed FMS. In the first two cases, MTQE would advise the translator to post-edit rather than translate from scratch, and in both cases it is the correct course of action judging by the observed FMS. In example (2), the word "según" is functionally equivalent to the phrase "en función de", and the change itself is a stylistic choice on behalf of the post-editor. In this example, the observed FMS would have been closer to the predicted FMS if not for this choice. In the case of Example (4), the predicted score seems to differ wildly from the observed score, with the predicted score suggesting to the translator that they should post-edit, while the observed FMS suggests they should translate from scratch. On closer examination, we see that the Spanish MT translation has all the words in the wrong place, and although the sentence is grammatically correct, the structure itself is wrong. Therefore, while the predicted score is too high, the observed FMS does not properly reflect the PE effort either.

(1) Sample Prediction 1 - Good QE

a. Source: To Navigate the Marking Menu Selections.

b. MT: Para navegar por las selecciones del menú de comandos frecuentes

c. PE: Para desplazarse por las selecciones del menú de comandos frecuentes

d. FMS: 88.661 (Predicted)/88.000 (Observed)

(2) Sample Prediction 2 - Good QE

a. Source: Stylizes each point based on the normal of the point

b. MT: Stylizes cada punto según la normal del punto.

c. REF: Aplica un estilo a cada punto en función de la normal del punto.

d. FMS: 87.03 (Predicted)/85 (Observed)

(3) Sample Prediction 3 - Good QE

a. Source: Create better buildings with intelligent 3D model–based design.

b. MT: Crear mejores edificios con modelos 3D avanzados basados en diseño.

c. REF: Cree mejores construcciones gracias al diseño basado en modelos 3D. 0 54 59.9523

d. FMS: 59.95 (Predicted)/54 (Observed)

(4) Sample Prediction 4 - Bad QE

a. Source: For example, intensity, normal, abstractname or classification data may not be available with a point cloud.

b. MT: Por ejemplo, la intensidad, normal o datos de clasificación pueden no estar disponibles con una nube de puntos.

c. REF: Por ejemplo, es posible que los datos abstractname de intensidad, normales o clasificación no estén disponibles en una nube de puntos

d. FMS: 90.88 (Predicted)/55 (Observed)

5.4 The User Study

5.4.1 PET: Post-Editing Tool

For our study we used PET[15] (Aziz et al., 2012) as our post-editing tool. Like other CAT tools, PET provides an easy to use user interface which facilitates both translating and post-editing. In addition, the tool records a number of statistics such as the keystrokes pressed and the time needed to perform the translation, which are very relevant for this research. Even though PET is not normally used in a real-world professional post-editing situation, it is ideal for acquiring data like the ones collected for our research. The tool is open-source and written in Java, which

[15] `http://www.clg.wlv.ac.uk/projects/PET/`

allowed us to easily modify the code to incorporate the traffic light system described in Section 5.4. While other tools such as SDL Trados Studio[16] or MemoQ[17] would have been preferred by the translators due to familiarity, these tools did not allow the same kind of malleability and customisation as PET, which allowed us access to the source code in order to edit in our traffic lights.

Figure 5.1 shows a screenshot of the out-of-the-box unedited PET interface. The bottom yellow box shows the current sentence to be translated on the left and the MT suggestion on the right. The top yellow box is for the translator to edit.

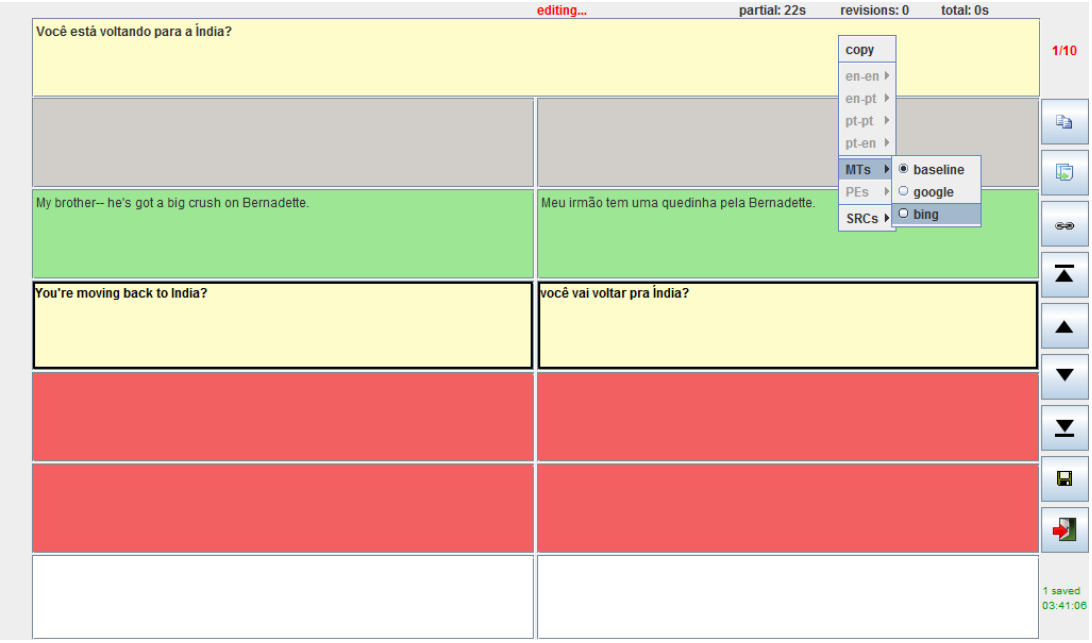


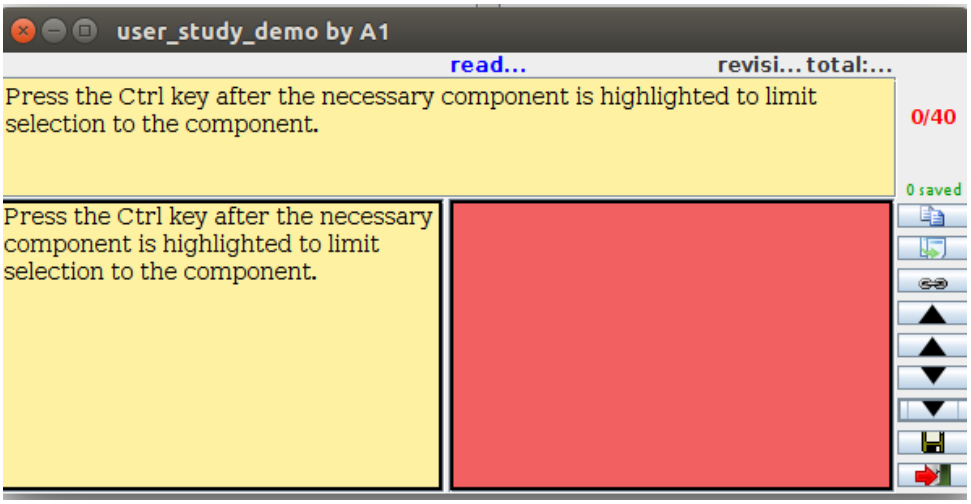
Figure 5.1: A Screenshot of PET out of the box

5.4.2 Settings of the User Study

We designed a traffic light system using PET to present translators with three different categories of sentences. The translators were presented with sentences (in English) to translate into Spanish. Some of these sentences had MT suggestions

[16] http://www.sdl.com/solution/language/translation-productivity/trados-studio/
[17] https://www.memoq.com/en/

Figure 5.2: Translate from scratch



to post-edit. All sentences were presented with one of four traffic lights, detailed in Table 5.2). A light yellow background indicated that a translator must translate the given sentence from scratch (in this case, the translator will not be given an MT translated sentence to post-edit). A light blue background indicated that a machine translation is available, however, no MTQE information is forthcoming, and therefore the translators must decide for themselves whether to translate from scratch, or to post-edit. A light green background indicated that the MTQE system strongly suggests that the translator post-edit the sentence. This meant that the MTQE system has predicted a fuzzy match score of 75 or more. Finally, a light red background indicates that the MTQE system strongly suggests that the translator translates the sentence from scratch. This meant that the MTQE system has predicted a fuzzy match score of less than 75.

Figures 5.2 and 5.3 show how the colour coding system was displayed to the translators.

Figure 5.3: Post-edit without MTQE

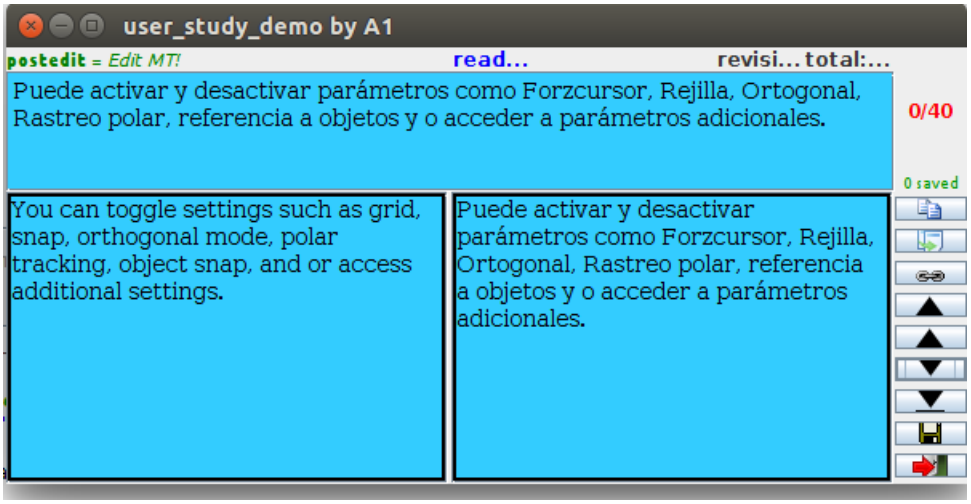


Table 5.2: A summary of the traffic light system.

Light yellow	No MT suggestion.
Light blue	No Quality Estimation, the Post-editor must make up their own mind.
Light green	The Quality Estimation recommends post-editing rather than translating from scratch.
Light red	The Quality Estimation recommends translating from scratch rather than post-editing.

The categories above were determined on the basis of the predicted MTQE scores. We know that some of these scores are not correct. For this reason a different way of looking at the data is to divide it into four categories: *No MT*, *No QE*, *Good QE* and *Bad QE*. These categories are summarised in table 5.3. In the first category, *"No MT"*, the translator was not presented with an MT suggestion to post-edit, and must therefore translate from scratch. In the *"NO QE"* category, the translator was provided with a MT suggestion, however, there is no MTQE suggestion, so they must make up their own mind whether or not to post-edit. The *"Good QE"* and *"Bad*

QE" categories both provided a MTQE suggestion to the translator, prompting them either to post-edit or translate from scratch. However, the "Good QE" category was made up of sentences for which the MTQE system predicted a FM score close to the observed FM score (within 10% of the observed score). The *"Bad QE"* category consisted of sentences where the MTQE system did not perform as accurately, and suggested a score that diverged from the observed FM score. The user was not told which sentences were *"Good QE"* and which are *"Bad QE"*. This data is known only to the researchers. We hoped that this categorisation would show us the effect of good and accurate MTQE specifically on the time and effort of the post-editor.

Table 5.3: Data Categorisation

Label	Description
NO MT	No MT suggestion.
NO QE	MT Suggestion but no MTQE suggestion
Good QE	MTQE suggestion within 10% of observed FM score
Bad QE	MTQE suggestion more than 10% different to observed FM score

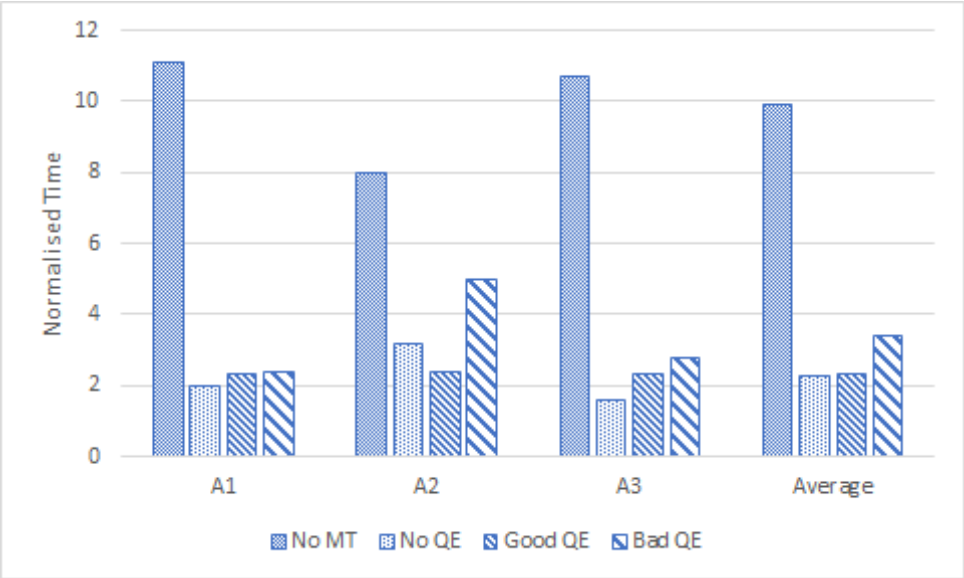
5.4.3 The Pilot Study

We performed a pilot study with 3 non-professional translators who are native speakers of Spanish. These translators were asked to look at a subset of 40 sentences from the full dataset. All 3 translators are native speakers of Spanish with at least a working proficiency of English. Despite their years of translating experience, none of the translators had any experience with technical texts and manuals. None of the translators were familiar with PET before participating in the experiment, and were not paid for their time. The purpose of the pilot was to identify potential problems the translators might run into, and we did not expect usable results from such a small dataset.

We extracted the post-editing times and keystrokes for all 3 translators. We normalised these results by dividing the time by the number of words in the source sentence. We also discarded one sentence, sentence number 4, because the post-editing time exceeded 9000 seconds. In cases where a translator skipped a sentence, we discarded their statistics as well.

Figure 5.4 shows the time in seconds each annotator spent on a given category of sentences (*No MT*, *No QE*, *Good QE* and *Bad QE*). As we expected, the sentences that needed to be translated from scratch took the most time across all annotators, showing that even among translators with no previous post-editing experience, MT can considerably boost translator efficiency. However, on first glance it does not seem that the inclusion of MTQE improves efficiency to any significant degree. This may be because only half of the MTQE included is accurate.

Figure 5.4: Number of seconds per word spent translating/post-editing per category



We also took a look at the results broken down by quality of MTQE rather than category. However, the difference between *No QE* and *Good QE* was small. It remained difficult to draw conclusions from such a small dataset. While the use of *Good QE* seemed to be overall preferable to *Bad QE*, there was no statistically significant or

consistent improvement in translator efficiency between using MTQE or simply giving the translator the choice to post-edit or translate. This could be due to several factors. The translators themselves were not familiar with post-editing, which may have affected the results. Furthermore, Autodesk's MT system seems to perform pretty well, leading to very little editing of the suggested machine translation. Finally, we used only 40 total sentences in the pilot study (10 per category) which may be the contributing factor for our inconclusive results. Despite these inconclusive results, the pilot study fulfilled its purpose to test the experimental setting and confirm that the instructions provided were clear and no unforeseen issues arose.

5.4.4 The Full Study

For the purpose of the full study, our total sentences number 260 (about 3,000 words). This number represents a day's work for the average professional translator. This allows us to emulate a real-world setting by asking the translators to complete the task in one day. We divided the annotated data into the 4 categories according to Table 5.3 and randomly selected 65 sentences from each of the 4 categories, to get a total of 160 sentences.

We enlisted the help of 4 professional translators with several years' English-Spanish translating experience. The years of experience varied greatly, between 3 and 14 years experience. All 4 translators had some experience with Post-Editing tools. All 4 translators are native speakers of Spanish with a working proficiency of English and were asked to fill out questionnaires before and after completing the tasks with the aim of gathering information about their background and their experience while performing the task. Table 5.4 summarises the translator details.

Table 5.4: Translator Summaries

Translator	C	M	V	S
Experience in technical domains (years)	14	6	3	6
Experience as a professional translator (years)	14	6	3	9
Experience with post-editing tools (years)	2	4	3	1
Opinion of Computer-Assisted Translation tools	+	+	+	+
Opinion of post-editing tasks	-	+	+	+

While all translators had some experience with post-editing tools, none of the translators were familiar with PET before participating in the experiment. To overcome this issue, together with the instructions to carry out the task for the experiment, we also provided them with a short user manual of the tool with screenshots aiming at familiarising them with the interface prior to the task itself. All translators were paid for their time and were asked to complete the task over the course of a day, in order to simulate the real-world experience.

5.5 Results and Analysis

PET records all the operations carried out by the translators. These operations are saved in an XML file which in turn can be used to analyse the translation process. This section is structured as follows. It first presents an analysis of the productivity of the translators measured using time and number of keystrokes (Section 5.5.1). The effect of good and bad MTQE information on the post-editing is analysed in Section 5.5.2, whilst the effect of the FMS on the post-editing information is presented in Section 5.5.3. The same sentences were shown to translators. We compare the resulting translations in Section 5.5.4. The section finishes with discussion of the translators' feedback.

5.5.1 Analysis of Productivity

We extracted the post-editing times and keystrokes for all 4 translators. We then normalised these results by dividing each by the number of tokens in the source sentence in order to compare sentences of different lengths. In cases where a translator skipped a sentence, we discarded the statistics for that sentence. In such cases, we discarded the sentence data for all translators, in order to ensure the results remained comparable. In total, we discarded 4 sentences this way.

Figure 5.5 shows the time, measured in seconds per word, that each translator spent on a given type of task translating from scratch – "*No MT*"; raw post-editing – "*No QE*"; and post-editing with MTQE information – "*QE*"). Each translator is identified by a letter. In addition, we provide the average for all four translators. As we expected, the sentences that needed to be translated from scratch took the most time across all translators, even without taking into account the quality of the QE. This seems to suggest that MT can considerably boost translator efficiency. The sentences that needed to be translated from scratch took the most time across all annotators, MTQE can considerably boost translator efficiency. This in itself is not an unexpected result, as MT is widely used in the Post-Editing workflow to reduce the translating effort. However, the more interesting results, for us, are the differences between post-editing with MTQE and without MTQE. We take a closer at the bars marked "*No QE*" and those marked "*QE*". The normalised (by sentence length) number of seconds drops from an average of 2.9 seconds to 2.4. This indicates that MTQE cuts post-editing time by an average of 0.5 seconds per word on average. Individually, however, this drop does not account equally over each translator. For two of the translators, the change is not significant. We will discuss the possible causes for this in Section 5.5.5.

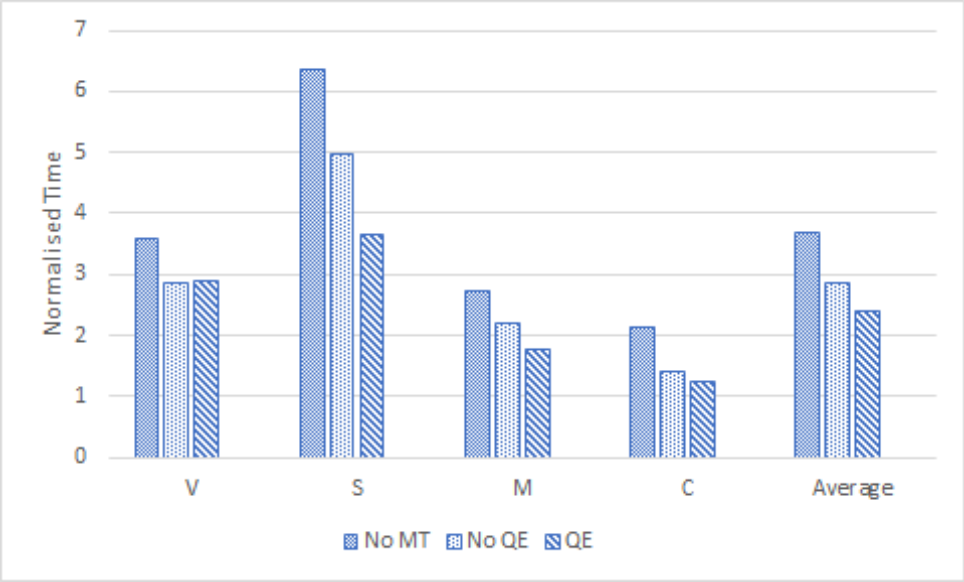


Figure 5.5: Number of seconds per word spent translating/post-editing

Figure 5.6 analyses the activity of translators from the point of view of the number of keystrokes per word, based on the type of task (translating from scratch – "*No MT*"; raw post-editing – "*No QE*"; and post-editing with MTQE information – "*QE*"). This helps us measure the physical effort in addition to the temporal one. As expected, the number of keystrokes used in the "*No QE*" and "*QE*" conditions in Figure 5.6 is clearly lower than the number of keystrokes used when translating from scratch. The same carries over when measuring the difference in keystrokes for post-editing with and without MTQE. The average number of keystrokes drops from 3.67 for "*No QE*" to 2.25 for "*QE*". This suggests that the inclusion of MTQE cuts post-editing effort by 0.4 keystrokes per word.

The reduction of the number of keystrokes between the setting where no automatic translation is provided and those where a translation was available is much greater than the reduction of the time between the same settings. This is to be expected given that even when a good automatic translation is available, translators need to spend time to read it and assess its quality.

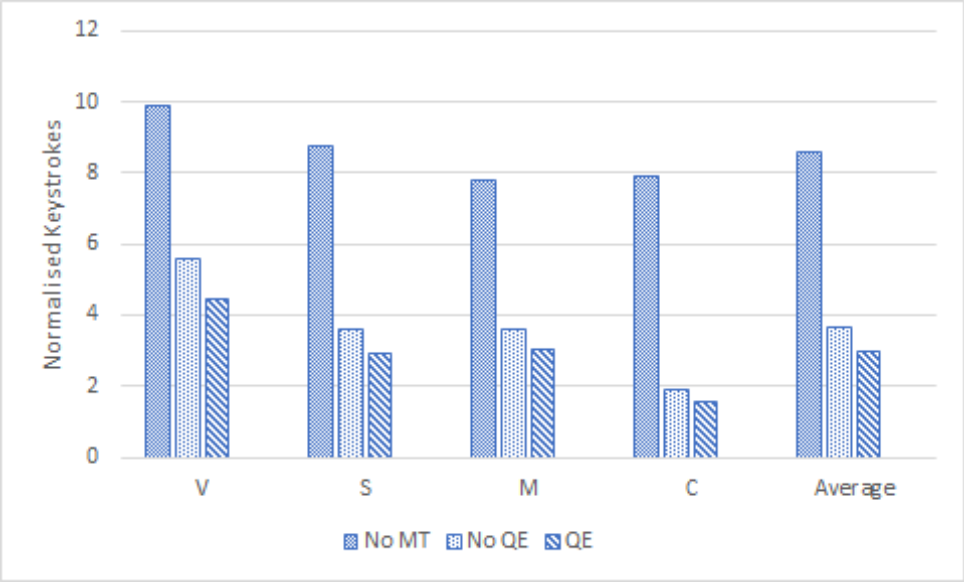


Figure 5.6: Number of keystrokes per word spent translating/post-editing

5.5.2 The Effect of *"Good QE"* vs *"Bad QE"* on Post-Editing

Our results so far do not fully reflect a real-world setting, as we have artificially increased the sentences with low quality MTQE by selecting 50% of the sentences with the *"Bad QE"* category. However, in our test, we observed an 85% accuracy of MTQE labels. Therefore, a fully random selection would have resulted in about 15% of the sentences falling into the "*Bad QE*" category instead. However, we wanted to study the impact that MTQE accuracy can have on post-editing effort. Table 5.5 shows the number of sentences in each category once divided by quality of MTQE and after the non-viable segments were discarded.

Table 5.5: Data Categorisation and Number of Sentences by Quality of MTQE

Label	Description	N
NO PE	No MT suggestion.	65
NO QE	MT Suggestion but no QE suggestion	63
Good QE	MTQE suggestion within 10% of observed FMS	63
Bad QE	MTQE suggestion more than 10% different to observed FMS	65

Figure 5.7 shows the time, measured in seconds per word, that each translator spent on a given type of task depending on the accuracy of the MTQE. Each translator is identified by a letter. In addition, we provide the average for all four translators. We also provide the results for the *"No QE"* category, for comparison. In terms of time spent post-editing, there is no significant difference between the "*Good QE*" and "*Bad QE*" categories on average. Only one translator showed any improvement between the two categories, this is Translator "S". For both Translator "C" and Translator "M", the results show no significant difference regardless of the accuracy of QE. For Translator "V", there was an increase in both time and effort between "*Good QE*" and "*Bad QE*". Figure 5.8 shows the same breakdown for keystrokes instead of time. Once again, on average there is little to no difference between "Good QE" and "Bad QE". Here Translator "V" seems to reuse segments tagged as "*Bad QE*" more than others, as does Translator "C", though to a much less significant extent. These erratic and strange results led us to take a closer look at the range of FMS for these individual sentences.

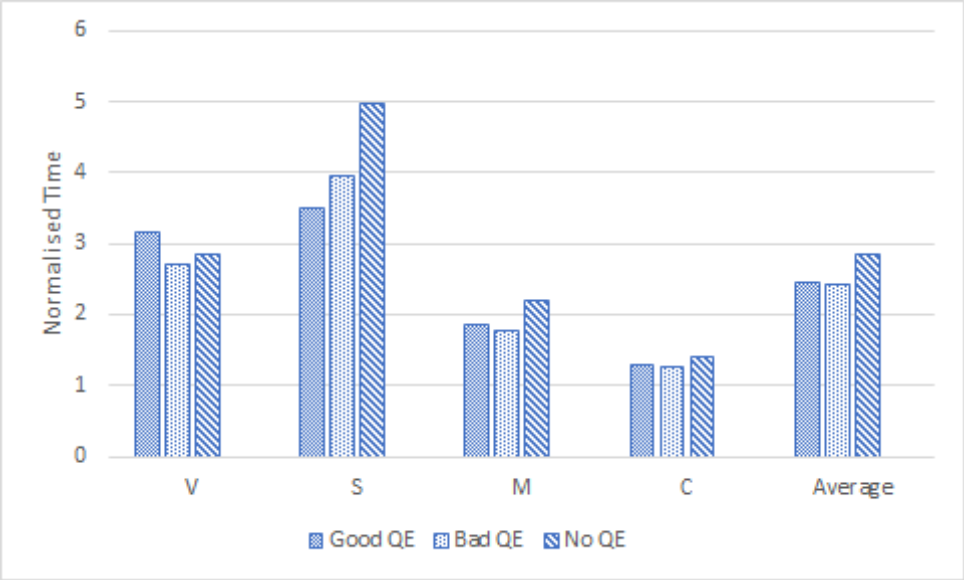


Figure 5.7: Number of seconds per word spent translating/post-editing

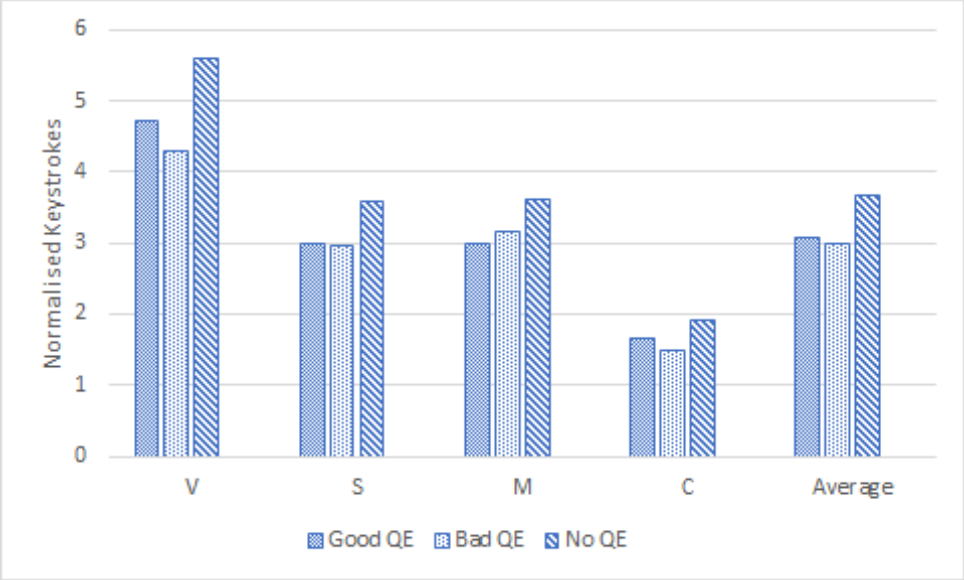


Figure 5.8: Number of keystrokes per word spent translating/post-editing

5.5.3 The Effect of the Fuzzy Match Scores on Post-Editing Effort

Table 5.6 shows a breakdown of the number of sentences by FMS range and the labels shown to translators (*Post-edit* or *Translate*). Of the sentences labelled as

post-edit, about two thirds (58 out of 82) are labelled correctly (i.e. the label predicted by the MTQE system matches that extracted from the the observed FMS score) and only 24 are labelled incorrectly. Conversely, among the 46 sentences labelled translate, the opposite is true. 33 sentences are labelled incorrectly, and only 13 are labelled correctly. According to the findings of Parra Escartín and Arcedillo (2015b), sentences with FMS <=75 are not useful for post-editing. This suggests that the high number of sentences in that range may have impacted the number of keystrokes observed in the previous section.

Table 5.6: Number of Sentences by Range

FM Range	**<=75**	**(75-100]**	**Total**
Post-edit	40	42	82
Translate	21	25	46

Therefore, we divide the sentences in our data into four categories:

- *GoodQE Translate*: (21 sentences) The observed FM score is <75, and the user is given a red light.
- *GoodQE Post-edit*: (40 sentences) The observed FM score is >75, and the user is given a green light.
- *BadQE Translate*: (42 sentences) The observed FM score is >75, and the user is given a red light.
- *BadQE Post-edit*: (25 sentences) The observed FM score is <75, and the user is given a green light.

In this section, we organised the sentences by their observed Fuzzy Match scores. The "Good QE" and "Bad QE" categories indicate whether the translator was given the correct or incorrect prompt. For comparison, we also include the "*No QE*" category as a control group. We also include the average over all 4 translators for each

category. Figure 5.11 shows the normalised time for each translator spent on sentences with FMS scores <= 75. Here a correct label (*"Good QE"*) would indicate translating from scratch, and an incorrect label (*"Bad QE"*) would indicate attempting to post-edit. On average, it seems that in the case of these particular sentences, attempting to post-edit still cut down the time and effort (overall), despite the low quality of the machine translation. This is especially pronounced for Translator "V" and Translator "S". This might indicate that these translators were more likely to follow the traffic lights' suggestion.

Things are much less defined in the case of translations with higher FM scores, however. Here, the MTQE suggestions have a less significant impact on translator time and effort. Figure 5.9 shows the normalised time spent post-editing sentences with a FMS > 75. With the exception of Translator "S", none of the translators show a significant difference in time between "Good QE" and "Bad QE". The same can be observed for effort in Figure 5.10.

Despite the indication that the quality of MTQE does not strongly affect the time and effort spent post-editing, it does seem that ' *"Good QE"* still improves over *"No QE"*. This is especially true for sentences with a FMS <= 75. This seems to suggest that MTQE is most helpful in cases of low MT quality, where the decision to post-edit or translate from scratch is difficult. This is also in line with Turchi et al. (2015), who found that the improvements in efficiency are only statistically significant for instances where HTER>0.1.

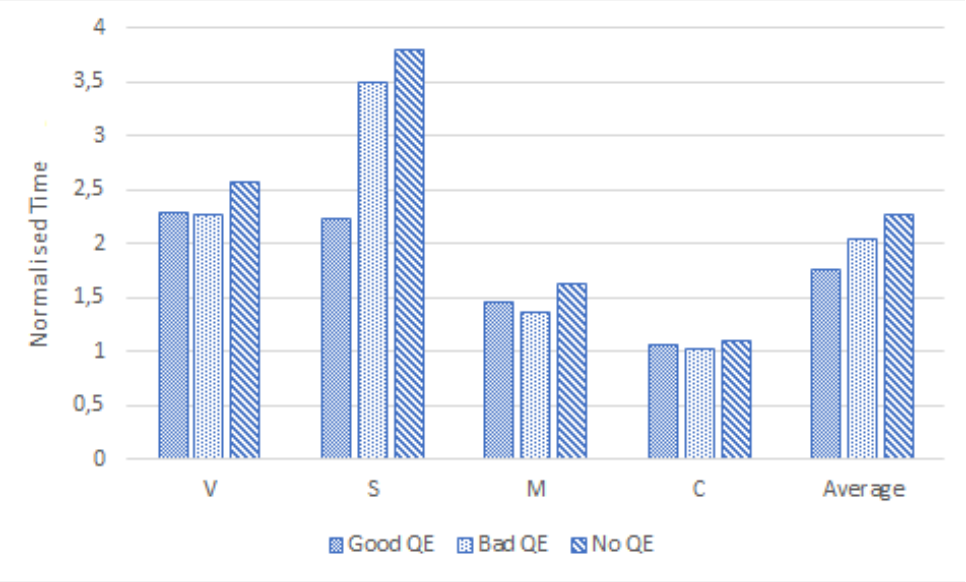


Figure 5.9: Normalised Time Sentences with FMS scores > 75

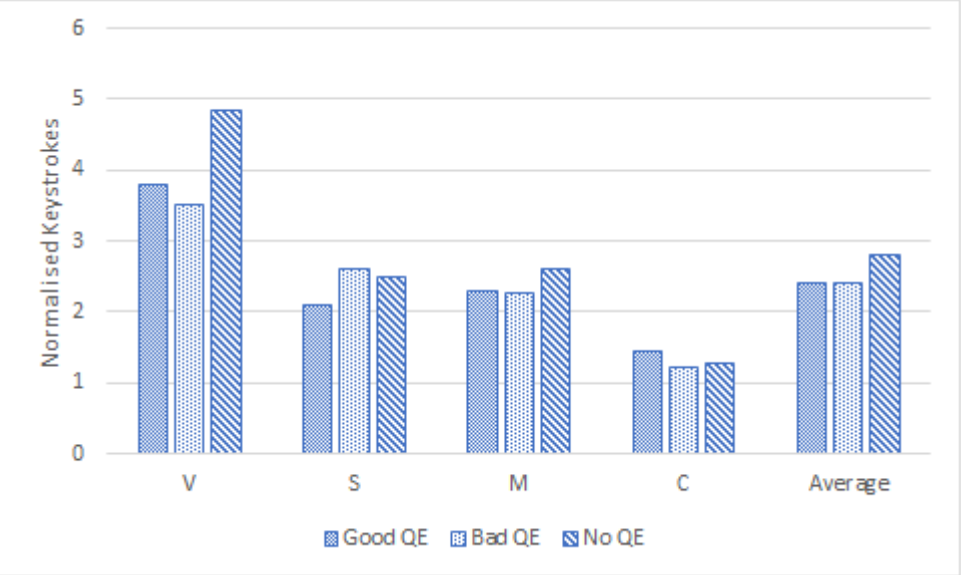


Figure 5.10: Normalised Keystrokes Sentences with FMS scores > 75

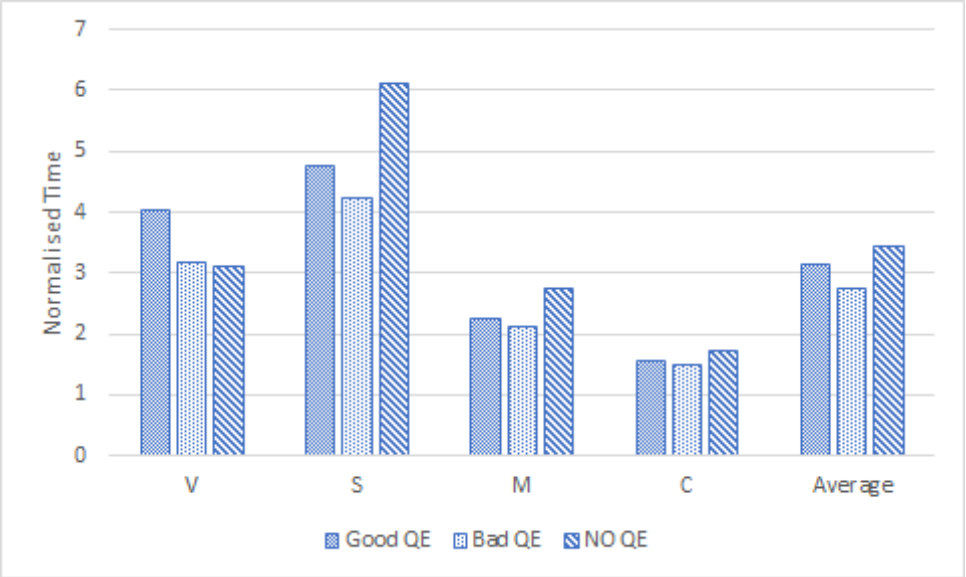


Figure 5.11: Normalised Time for Sentences with FMS scores <= 75

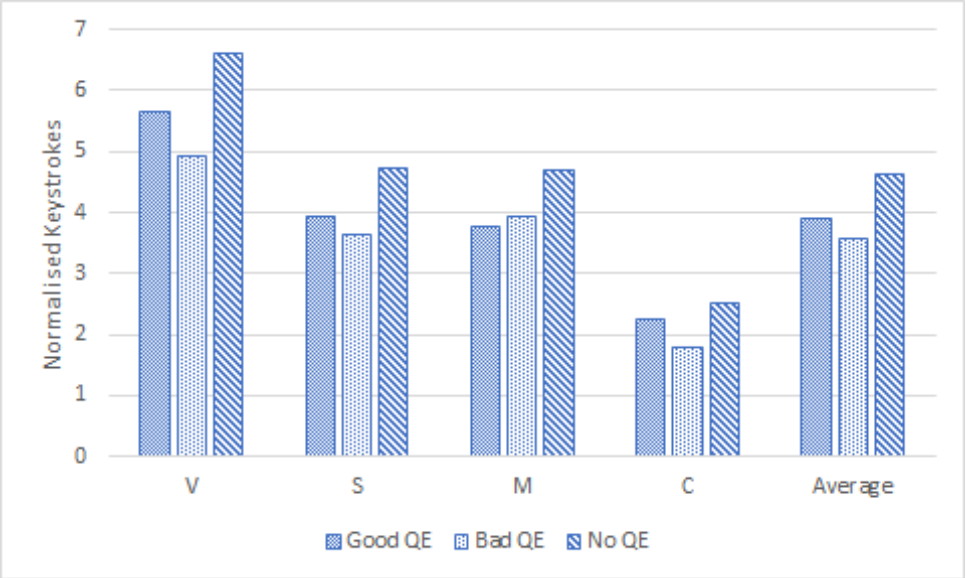


Figure 5.12: Normalised Keystrokes for Sentences with FMS scores <= 75

5.5.4 Quality of Translation

In order to find out whether there are differences between the resulting translations, we take a look at the FMS scores of the translators' sentences, comparing them to the post-edited reference provided by Autodesk. In the context of this paper, we

report the results using FMS for comparison in Table 5.13. In addition to FMS, we also take a look at the BLEU scores in Table 5.14.

We find that despite their varying levels of experience, all 4 translators achieved fairly high scores. We found that the FMS scores vary the most for the "*No MT*" category, with an average standard deviation of 7.6. This result is expected, as we expect the results to vary the most when the translators do not have an MT suggestion as a starting point. Furthermore, they are not Autodesk usual translators and are not familiarised with the terminology and style required by Autodesk. As their MT engine is deployed in-house, it is expected that it mimics the style of their translators and uses the right terminology.

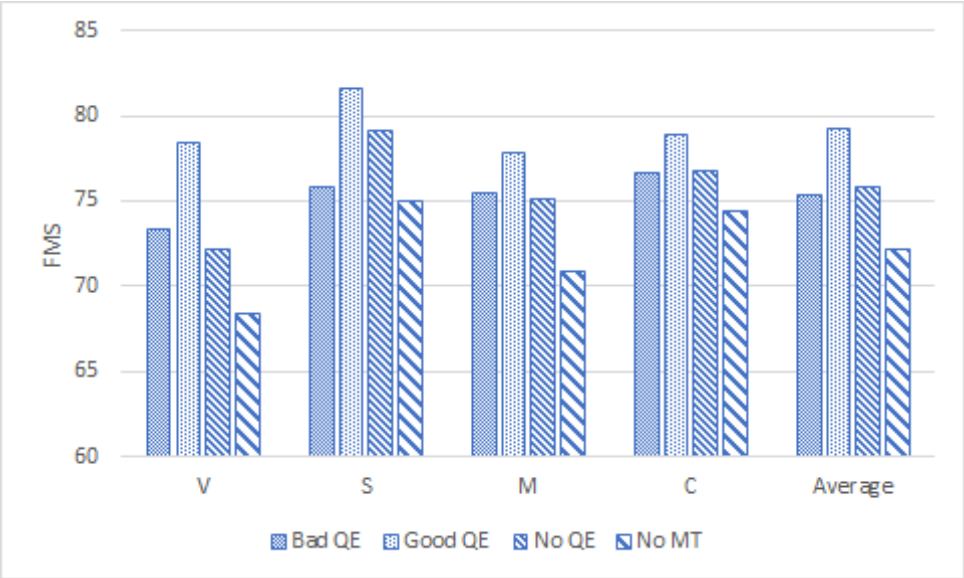


Figure 5.13: FMS scores for post-edited sentences

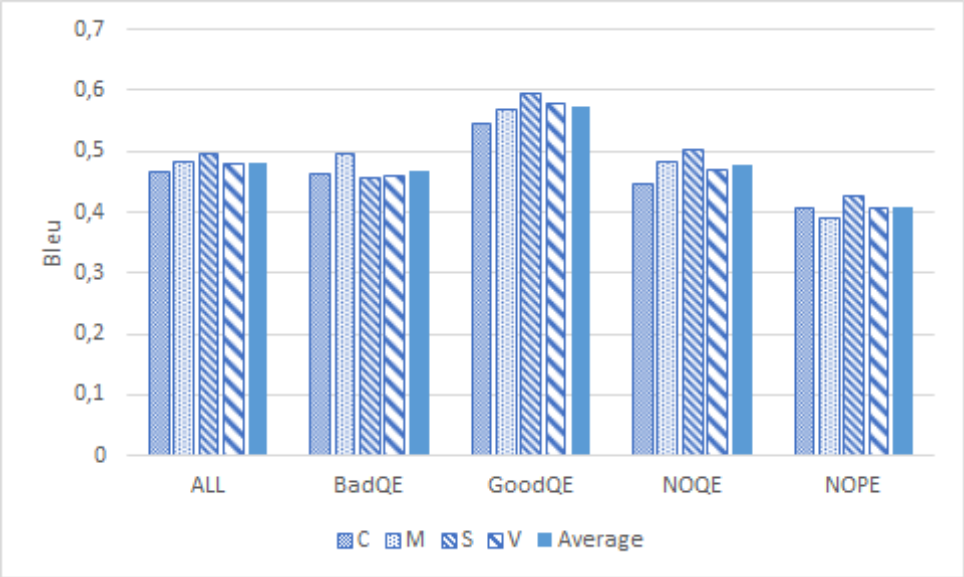


Figure 5.14: BLEU scores for post-edited sentences

5.5.5 Analysis of the Translators

In Table 5.4, we summarised the translators' years of experience both as professional translators and post-editors. When we compare these findings to the results in Figure 5.5, we find a negative correlation between the years of experience and the time spent both translating and post-editing. The exception is Translator "S", who despite 9 years as a professional translator, spends over twice as much time per word as the rest of the translators. This could reflect the lack of experience that Translator "S" possesses in terms of post-editing tools. Despite the discrepancy in time, the keystrokes of Translator "S" do not seem to vary that much from the rest of the translators in terms of effort (keystrokes in Figure 5.6). This lends more weight to the theory that the time spent getting familiar with the PET tool is responsible for the additional time observed in the results.

We asked all the translators to fill out questionnaires before and after the task in order to gain a more first-hand perspective of translators and post-editing tools. Responses suggest that while all four translators approved of the MT suggestions,

all found the Post-editing Tool difficult to navigate, which may have affected both their results and opinions of Quality Estimation. Despite the results above, three of the translators answered that they did not find Quality Estimation helpful. One translator disagreed, saying that they liked getting a first impression via the traffic lights system. Three out of the four translators claimed that the MT suggestions were helpful, while one insisted that they were better off translating from scratch. Despite their impressions, the results show that the translators did benefit from MTQE. Translator "S" especially benefits from MTQE information, cutting down his time from 4.9 to 3.7 when MTQE information was included. And while translator "V" did not gain much in terms of time, MTQE-informed post-editing cut down the effort (in terms of keystrokes) by 1 keystroke per word.

Table 5.7: Professional Translators and their Opinions after the Study

Translator Opinion	C	M	V	S
Professionalism of Task	Yes	No	Yes	No
MT Quality	Good	Good	Good	Good
Usefulness of QE	Bad	Bad	Good	Bad
Accuracy of QE	Bad	Bad	Good	Bad
Opinion of Quality Estimation	Negative	Negative	Positive	Negative

5.6 Conclusion

In this chapter, we set out to answer the research question RQ2:

RQ2 To what extent does the use of quality estimation tools affect the efficiency of the translation workflow?

We designed and implemented a user study to investigate the impact of using MTQE information in the post-editing workflow. To achieve this, we ran a study us-

ing 260 sentences from the Autodesk Post-editing Parallel Corpus, annotated for fuzzy matches using QuEst++ with semantic features. The semantically enhanced QuEst++ system, trained and tuned on Autodesk data, performed quite well for our test set. The observed MAE for the Autodesk data corpus is of 9.5 and we obtain an accuracy of 85% for the labels *Translate* and *Post-edit*. Further to these findings, we conducted a user study to investigate the impact of using the MTQE information in the post-editing workflow. After a pilot study with only 40 sentences to test our setup, we invited 4 professional translators to take part in a post-editing/translation task, using a traffic lights system to provide MTQE information. The translators were asked to use PET to use PET as their translation environment. We chose PET because it records all the editing operations performed during the translation. All four translators were paid for their work. Our results show that MTQE, especially good and accurate MTQE, is vital to the efficiency of the translation workflow, and can cut translating time and effort significantly. This seems to contradict the findings of Teixeira and O'Brien (2017), who find no significant difference in time or effort for post-editing for MTQE. Translator feedback still seems quite negative in spite of this improvement, however, which suggests a better post-editing tool might be required to win over the hearts and minds of translators. In conclusion, we have adequately answered the question we asked in RQ2, as we have shown that MTQE can help professional translators in the postediting process, in a real-world setting. We have shown that MTQE helps in assessing which sentences are worth postediting and which should be translated from scratch, and helps cut down the time and effort of the posteditor.

Chapter 6

Other NLP Evaluation Applications for STS

6.1 Introduction

In the previous chapters, we showed how STS is useful for MTQE. However, STS can be useful to a large number of NLP applications, such as information retrieval, text summarisation and question answering, as we previously discussed in Section 2.2. This chapter explores how STS can help with evaluating text simplification and translation memory matching and retrieval, and attempts to answer the third research question, which we divided into two subquestions: RQ3.1 and RQ3.2.

[RQ3] Can we explore the applications of Semantic Textual Similarity further:

RQ3.1: in automatic evaluation of simplified text?

RQ3.2: in translation memory matching and retrieval?

This chapter is split into two parts. In Section 6.2, we propose and investigate the use of STS tools for the automatic evaluation of simplified sentences. We augment

the STS system described in Section 3.4 with features that measure the simplicity and grammaticality of sentences in addition to meaning preservation, in order to fully encompass the needs of the simplification system's audience. We apply the machine learning techniques we used in previous chapters to solve the problem of text simplification evaluation. Our classifiers and feature sets show good results, especially in the realm of meaning preservation.

In Section 6.3, we attempt to answer RQ3.2 by testing the STS system against a basic edit distance for translation memory matching. Translation memories are computer aided translation tools that work by retrieving previously translated segments, or closely matching translated segments. These segments are usually stored in a database that is updated as users continue to translate and edit matches. The larger the database, the more effective the translation memory is. In previous years, translation memories (TM) have used simple edit distance and fuzzy match methods to retrieve matches. These metrics largely rely on surface form, and can miss sentences that may be paraphrased (Gupta et al., 2015b). Therefore, a more sophisticated TM matching metric could theoretically improve the performance and usefulness of a TM. We test the performance of our STS system (c.f. Section 3.4) on TM matching and compare it to the basic edit distance metric.

6.2 Automatic Text Simplification (ATS)

ATS tools attempt to improve the simplicity and readability of texts while conserving the meaning of the original. ATS tools transform complex sentences into simpler, more readable ones. Longer sentences are reduced and broken up, and made simpler for readers with cognitive impairments, dyslexia, aphasia, autism or similar reading difficulties. These tools are usually evaluated by measuring reading speed and comprehension by target users (Rello et al., 2013; Fajardo et al., 2014) and by human annotators who assess sentences in terms of grammaticality, meaning preser-

vation, and simplicity (Woodsend and Lapata, 2011). As with human evaluation of machine translation output, this remains costly and time consuming. The problem of simplicity introduces further complications than that of machine translation evaluation, as annotators must also be familiar with the specific requirements of the target audience. Automatic evaluation can be both faster and more consistent and can enable an assessment and comparison of ATS systems on a much larger scale. In this section we explore the use of STS tools to evaluate the output of automatic text simplification systems. Section 6.2.1 presents a short overview of ATS and previous methods used to evaluate them. Section 6.2.2 presents the data used in our experiments. Section 6.2.4 presents our approach to evaluating ATS output using STS methods. Section 6.2.5 presents our results and findings. Finally, Section 6.2.6 presents our conclusions.

6.2.1 Background

Research into ATS explored a range of different approaches that are largely independent and methodologically distinct. These methods range from rule-based lexical to syntactic simplifications, explanation generation, and statistical machine translation (Shardlow, 2014). In this research we focus on ATS evaluation and not on developing our own simplification method. Therefore, we only give a brief overview of the various methods available.

A number of rule-based approaches have been developed since 1990. In these approaches, language processing tools such as part of speech taggers are used to match patterns and apply the relevant rules accordingly. Rules are often handcrafted, and are used to split sentences' sub-clauses or resolve pronominal anaphora to make sentences less complex and easier to follow (Siddharthan and Mandya, 2014; Rennes and Jönsson, 2015; Ferrés et al., 2015; Evans and Orăsan, 2019).

More recently and with the availability of more relevant resources (such as Simple

Wikipedia, a simplified version of Wikipedia), data-driven approaches have found their way into the field of ATS. Specia (2010) use the standard Phrase Based statistical machine translation (PBSMT) model provided by the Moses toolkit (Koehn et al., 2007) to translate from "original" Brazilian Portuguese to "simple" Brazilian Portuguese. The dataset contained manual simplifications aimed at people with low literacy levels. By treating simplification as a translation problem, the original Brazilian Portuguese sentences as the source and the simpler version as the target, the authors were able to train a monolingual "translation" system. The system performed well, and despite the system's tendency to be overcautious in simplifying sentences, was able to achieve a BLEU score of 60.75, which is usually considered a good score when judging a translation system. The authors also perform some manual evaluation to get a better idea of the system's performance. The 20 randomly selected sentences received an average score of 2.5 out of 3 for fluency and adequacy, and 2.35 out of 3 for simplicity.

Coster and Kauchak (2011) use a similar approach for English, extending the PBSMT system by adding phrasal deletion to the probabilistic translation model. The authors train their system on a much larger corpus (124k aligned sentences), using English Wikipedia as the source language and Simple English Wikipedia as the target language. The authors report a BLEU score of 60.46 for this extended model. However, this proved to be only a small improvement over the baseline (59.37), which left all original sentences unchanged. These results motivated a deeper look into the influence of the quality of the training data on the effectiveness of PBSMT approaches (Štajner et al., 2015). The authors conduct upwards of 40 experiments on Original and Simple Wikipedia texts, controlling for size of the dataset and the extent of simplification between the original and simplified sentences. The authors show that PBSMT approaches sacrifice fluency when they increase simplicity. The authors conclude that the performance of these systems can be significantly improved with a more carefully selected training set.

This short and incomplete preview of the field of ATS gives us an inkling of the difficulty of the ATS task. As research into these methods continues, the need for reliable and efficient tools that evaluate these systems has become essential. While automatic evaluation and quality estimation have become a staple of machine translation evaluation, these methods have only recently made their way into the evaluation of text simplification. Štajner et al. (2014) investigated the application of widely used MT evaluation metrics to the output of text simplification systems. The authors posited that automatic MT evaluation shares many similarities with automatic text simplification, and therefore the same tools can be applied to both tasks. According to their research, many of these metrics show a strong correlation with human judgments when applied to the evaluation of automatic text simplification. Most significantly, BLEU (c.f. Section 2.3) shows the highest correlation with human judgement for grammaticality at 0.442 Pearson. Their study, however, has some limitations, focusing primarily on syntactic simplification which results in considerable content reduction.

In 2016, the Language Resources and Evaluation Conference hosted a shared task titled Quality Assessment for Text Simplification (QATS) (Štajner et al., 2016). The aim of this workshop was to investigate the use of automatic evaluation and quality estimation methods and their application to automatic text simplification. The shared task asked researchers to automatically assess the output of automatic text simplification and assign each sentence to one of three classes: "good", "ok" and "bad". These scores were also divided into 4 aspects: simplicity, meaning preservation, grammaticality, and an overall score. QATS provided two tracks: the constrained track, which allowed participants use of the dataset provided by the shared task organisers only, and the unconstrained track, which allowed participants to augment the training data freely. We participated in this shared task in the constrained track with a system that is described in Section 6.2.4.

6.2.2 Data

The QATS organisers provided the data for the constrained track. The dataset consisted of 631 sentences extracted from the news domain and from Wikipedia articles and simplified using the following systems:

- 224 sentences from EMM NewsBrief[18] were simplified using the EventSimplify TS system (Glavaš and Štajner, 2015).
- 119 sentences from Encyclopedia Brittanica were simplified using phrase-based machine translation (Štajner et al., 2015).
- 240 sentences from English Wikipedia translated using 3 different lexical simplification systems including EventSimplify (Glavaš and Štajner, 2015), a context-aware approach described in Biran et al. (2011) and a lexical simplifier described in Horn et al. (2014).

The sentences were then broken down into 505 for training and the remaining sentences for testing. More details about the data can be found in Štajner et al. (2016).

6.2.3 Sentence Evaluation

All these sets were manually evaluated for (G)rammaticality, (M)eaning Preservation and (S)implicity.

- Grammaticality: Measures fluency and grammatical correctness of the simplified sentence.
- Meaning Preservation: Indicates how much information is preserved after simplification and how much meaning is lost.

[18] `emm.newsbrief.eu/NewsBrief/clusteredition/en/latest.html`

- Simplicity: Indicates the readability of the sentence compared to the original more complex version.

Based on these three criteria, each sentence was given three different scores, plus an overall score calculated by averaging all the three other scores. The guidelines for annotation are given in Table 6.1.

Table 6.1: Classification Guide for Automatically Simplified Sentence Pairs

	Grammaticality	**Meaning**	**Simplicity**
Bad	ungrammatical	complete loss of meaning	difficult to understand
OK	grammatically incorrect but understandable	some meaning is preserved	somewhat difficult to understand
Good	grammatically correct	All meaning is preserved	Easy to understand

6.2.4 Our Approach

In order to capture all 3 aspects of grammaticality, meaning preservation and simplicity, we incorporate 3 sets of features totalling 36 features in total.

1 **Quality Estimation Features (17)**: In order to estimate the fluency and grammaticality of the text, we first calculate the 17 baseline features used in quality estimation tasks and described Section 2.3.4. For the purpose of these features, we treat the TS dataset as a machine translation set, and its evaluation as a MTQE problem. We treat the original sentence set as the source language and the simplified sentences as the target. These features are extracted using QuEst++, with out-of-the-box settings, using the packaged resources.

2 **Semantic Similarity Features (13)**: Similarly, we see the meaning preservation problem as a semantic similarity task. As our STS features aim to

capture how similar two sentences are, it follows that they would help decide how much meaning is preserved. Therefore we use the system described in Section 3.4, with the 13 STS features used in the majority of this research.

3 **Simplicity Features (6)**: Finally, to address simplicity, we exploit the features described in Yaneva and Evans (2015). They investigate readability, specifically where it relates to Autism Spectrum Disorders (Jordanova et al., 2013). According to the American Psychiatric Association, Autism Spectrum Disorder (ASD) is a developemental disorder of neural origin, characterised by impairment in communication. Individuals with ASD may struggle at semantic, syntactic and most of all, pragmatic levels of understanding. The authors analyse 33 readability indices and identify 6 predictors of autistic text comprehension. The authors also apply these indices to the evaluation of automatic text simplification and found them successful in discriminating between original and simplified versions of text. The authors identify a number of readability metrics that correlate with the simplicity measure. Readability metrics are formulae for evaluating the ease of reading for texts, usually based on the number of characters per word, word per sentences, and syllables per word. These metrics are used to estimate how difficult a text is to read, usually on a US grade level. Every index is a little bit different, emphasising different aspects of text complexity. Some focus on syllable counts while others look only at word and sentence lengths. We use the following readability measures:

- The Flesch-Kincaid Grade Level (Flesch, 1948)

  This index is interpreted as the reading grade level and is calculated using the equation described in Equation 6.1.

$$FKRA = (0.39ASL) + (11.8ASW) - 15.59 \tag{6.1}$$

  where ASL is average sentence length and ASW is the average number

of syllables per word.

- The Automated Readability Index (ARI) (Senter and Smith, 1967)

  Like the Flesch-Kincaid Grade Level, ARI produces an approximate grade level for the reader based on the equation in Equation 6.2.

$$ARI = 4.71(characters/word) + 0.5(words/sentences) - 21.43 \quad (6.2)$$

- The Coleman-Liau Index (Coleman, 1971)

  The Coleman-Liau Index (CLI) similarly builds on the characters and words in sentences to determine readability. It is defined in Equation 6.3.

$$CLI = 0.0588L - 0.295S - 15.8 \quad (6.3)$$

  L is the average number of characters per 100 words and S is the average number of sentences per 100 words.

- The Fog Index (Gunning, 1952)

  The FOG index is determined by selecting a paragraph of about 100 words and determining the average sentence length in the paragraph, and adding it to the number of complex words in the paragraph. This is further demonstrated by Equation 6.4.

$$FOG = 4[(words/sentences) + 100(complexwords/words)] \quad (6.4)$$

  where complex words are words with 3 or more syllables.

- SMOG Reading Ease (Mc Laughlin, 1969)

  SMOG Reading Ease (see Equation 6.5) is determined by randomly extracting 30 sentences from the text and calculating the square root of the number of complex words that appear in these sentences, and adding 3

to that number. As with the FOG Index, complex words are words with 3 or more syllables.

$$SMOGgrade = 3 + \sqrt{P} \tag{6.5}$$

where P is the number of words with 3 or more syllables in a sample of 30 sentences.

- Lix Reading Index (Bjornsson, 1968)

  The Lix Reading Index was optimised for Western European languages and is defined by Equation 6.6.

$$LIX = W/S + (Cx100)/A \tag{6.6}$$

where A represents number of words, S is the number of sentences, and C is the number of complex words, defined here as having more than 6 characters.

6.2.5 Results

We built a classification model which divided our test sentences into 3 different classes:

- Sentence pairs marked "Good" were given the class "2".
- Sentence pairs marked "OK" were given the class "1".
- Sentence pairs marked "Bad" were given the class "0".

We chose to run our system on the 505 sentences provided by the shared task organisers and not augment our training set with any further simplified sentences. We used LibSVM, and employed a three-way-classification system. We optimised

for the values of C and γ through a grid-search which uses a 5-fold cross-validation method, and all systems use an RBF kernel.

Results were calculated using different error and correlation techniques:

1 **Accuracy**: The percentage of sentences where the observed and the predicted class are the same

2 **Mean Absolute Error (MAE)**: The arithmetic mean of absolute differences between the observed and the predicted classes

3 **Mean Squared Error(MSE)**: The square root of the arithmetic mean of squared differences between the observed and the predicted classes

Results were then compared to the baseline results, calculated using the widely used and well-known automatic evaluation metrics BLEU, METEOR and TER (c.f. Section 2.3.2). We use the baselines provided by the shared task organisers for comparison. The metrics were used to calculate the sentence-based scores between the original and simplified sentences. The results in Tables 6.2, 6.3, 6.4 and 6.5 show our system's performance compared to these baselines as well as their overall rank in the shared task. Table 6.2 suggests that the strongest area our system performs in is meaning preservation, ranking third out of 22 systems and consistently outperforming the baseline metrics. This suggests that our STS features are strong predictors of meaning preservation. This is expected as we designed the STS features to measure the preservation of meaning between two sentences. Furthermore, it validates our decision to use STS as a means to evaluate Automatic Text Simplification methods. Our submission did not fare as well in other areas. While we outperform the baseline metrics on simplicity and rank in the top third, our predictors for grammaticality failed to outperform the baseline metrics. We also ranked last in grammaticality, compared to the 22 systems submitted to the shared task. However, despite our grammatical failures, our system's overall performance remains solid, not only out-

performing the baseline systems, but showing stronger predictive power than all but one of the other systems submitted. Our system ranked 2 out of 22 in overall performance.

Table 6.2: QATS results based on Meaning Preservation

	Accuracy(Rank)	**MAE(Rank)**	**MSE(Rank)**
STS System	63.49(9)	**20.63(3)**	**26.35(3)**
Baselines			
TER	**66.67(3)**	21.03(8)	28.10(8)
BLEU	65.08(7)	21.43(9)	27.59(6)
METEOR	61.90(13)	21.43(10)	31.25(14)

Table 6.3: QATS results based on Simplicity

	Accuracy(Rank)	**MAE(Rank)**	**MSE(Rank)**
STS System	44.44(13)	**28.17(7)**	44.19(14)
Baselines			
TER	8.10(16)	21.03(8)	46.23(21)
BLEU	38.10(17)	34.13(14)	45.77(18)
METEOR	35.71(19)	34.92(15)	47.47(22)

Table 6.4: QATS results based on Grammaticality

	Accuracy(Rank)	**MAE(Rank)**	**MSE(Rank)**
STS System	41.27(22)	30.16(22)	46.03(22)
Baselines			
TER	**66.67(3)**	21.43(13)	27.76(15)
BLEU	69.84(14)	21.43(12)	26.61(14)
METEOR	61.90(13)	24.21(17)	33.45(18)

Table 6.5: Overall QATS results

	Accuracy(Rank)	**MAE(Rank)**	**MSE(Rank)**
STS System	**50.79(2)**	**26.59(2)**	**33.11(2)**
Baselines			
TER	38.10(13)	40.87(17)	46.29(20)
BLEU	37.30(18)	41.27(19)	45.01(17)
METEOR	61.90(13)	21.43(10)	31.25(14)

6.2.6 Summary

Upon investigating the role of semantic textual similarity in the evaluation of automatic text simplification methods, we presented a total of 36 features, organised in 3 sets, each geared towards a different phenomenon: simplicity, meaning preservation and grammaticality. To address meaning preservation we use the features presented in Section 3.4, which are semantic features used to measure semantic textual similarity. The intuition is that more closely related sentences preserve the meaning better. To address grammaticality we use MTQE baseline features, treating automatic text simplification as a translation from "original" to "simple". Finally, to measure simplicity we use a number of readability ease metrics. We train a ML system which uses these features on 505 automatically simplified sentences, annotated manually for quality based on the three different criteria for quality.

On the Shared Task on Quality Assessment for Text Simplification (QATS), our classification systems, which used all 36 proposed features, ranked second overall among all participating systems and consistently outperformed the baseline for all types of quality measures. Our approach reported especially promising results for meaning preservation and simplicity, and for the overall quality measure, showing the potential of our approach being used for automatic evaluation of various ATS systems. The tool performed less impressively in determining grammaticality, performing even more poorly than the baseline metrics Bleu, Ter and Meteor. However, this is to be expected as our work focuses primarily on semantic similarity, and more care would be needed to identify features that select for grammaticality and fluency measures in the future. Furthermore, the features used to measure grammaticality are MTQE features, that have been designed and perfected to work in a translation context, and not in a simplification scenario, and that arguably have little to do with grammaticality.

6.3 Translation Memory Retrieval

A Translation Memory is a database of previously translated sentence pairs which acts as an aid to translators by enabling them to reuse translations. Previously translated segments can be retrieved from the database and reused or post-edited, cutting down on translation time and cost. TMs have become commonplace in the industry, and though methods for matching and retrieval can vary, they are still based on Edit Distance measures.

Most translation memories use some form of edit distance, or Levenshtein, to retrieve matches. Edit distance metrics show how similar two strings are by calculating the number of operations required to make one string identical to the other. However, several researchers have looked into improving these techniques by incorporating semantic information into the matching and retrieval process (Planas and Furuse, 2000; Macklovitch and Russell, 2000; Hodász and Pohl, 2005; Gupta et al., 2016). This section presents our approach to using STS as a tool for Translation Memory matching and retrieval. The rest of this chapter is organised as follows. In Section 6.3.1 we present some previous work into using STS in TM matching and retrieval. In Section 6.3.2 we detail our approach to using STS tools for TM matching. In Section 6.3.3 we test our approach and compare the results to those of a baseline edit distance. Finally, we present our conclusions in Section 6.3.4.

6.3.1 Background

In their publication, Planas and Furuse (2000) describe an algorithm for calculating semantic similarity between two (monolingual) segments to retrieve the best translation memory match. This algorithm, while quite similar to edit distance, does not allow for matches that require insertions. Furthermore, it looks at sentences as a group of layered segments, with each layer encapsulating a different level of infor-

mation for each sentence (such as surface form, Part of Speech, or lemma). This algorithm proved to be extremely efficient in retrieving TM matches and returned more usable results than a basic fuzzy match algorithm.

Gupta and Orăsan (2014) also argue for integrating semantic knowledge in the form of paraphrasing information in matching and retrieval. They show that surface form edit distance methods might miss matches that require little effort to post-edit. An edit distance method will miss paraphrases such as "the period laid down in article 4(3)" and "the duration set forth in article 4(3)", two sentences which have the same meaning but only a 57% Levenshtein score. The system they propose incorporates paraphrasing with edit distance and obtains significant improvement in translation memory retrieval and in translation. They use the paraphrase database (PPDB (Ganitkevitch et al., 2013)) to augment a TM with paraphrase matches for existing sentences. Instead of simply retrieving all matches (which the authors found highly inefficient), the paraphrases are included according to a classification system based on the number of words in the source and target sentences. They tested their method on the 2013 release of the DGT:TM (Steinberger et al., 2006) and achieved an improvement of 1.28% over the baseline systems (edit distance), retrieving 127 more matches. They also observed an increase of over 4 BLEU points, showing that their method improves matching as well as retrieval. In their follow-up paper (Gupta et al., 2016), the authors improve on this method, using greedy approximation techniques, and perform extensive human evaluation to measure the impact on translation efficiency. The authors conclude that their enhancements substantially improve TM matching and retrieval, with the ED match taking on average 33% more keystrokes and 10% more time to post-edit than the enhanced match. Like the previous work presented in this section, we attempt to improve TM matching by using additional semantic information. Our approach uses the STS tool described in Section 3.4 to retrieve matches and compares these validity of these matches to those retrieved using an ED metric.

6.3.2 Our Approach

We use the STS system described in Section 3.4, which was trained on the SICK dataset described in Section 4.5.3 to determine a STS score between 1 and 5 for a pair of sentences. We then generate our test set using a random selection of segments from the DGT-TM corpus described in Section 4.4.3. We select 500 sentences to be our test set. We also create a TM set from 5,000 randomly selected segments from the rest of the corpus. This TM set is the pool from which we will find our matches. There is no overlap between the sentences in the test set and the TM pool. For each sentence in the test set, we extract 5,000 STS scores, one for each segment in the TM pool. We then rank the sentences in the TM pool by these scores and retrieve the segment with the highest score. We repeat this approach for ED scores, and retrieve the segments from the TM pool with the highest ED score. We also retrieve the French translations of these segments, as we intend to use them for evaluation. After performing a test of these 500 sentences, we expand the experiment to encompass 2,500 test sentences and a TM set of 10,000 sentences. We denote the first experiment with 500 sentences as Test 1 and the expanded set of 2,500 sentences as Test 2. As our STS system works best with English sentences, we choose to use the English*to*French TM. The French sentences are only used for evaluation.

6.3.3 Results

We perform two types of evaluation on our approach: an automatic evaluation using Bleu and Meteor, and a manual analysis comparing the sentences selected by the STS systems versus that of the baseline ED approach.

The automatic evaluation compares the French translation of the retrieved TM match to the French translation of the original English sentence. These two sentences

are evaluated using popular MT metrics BLEU and METEOR and are detailed in Tables 6.6 and 6.7. The results of Test 1 show a marked improvement in BLEU and a slight improvement in METEOR. However, the same results do not hold true for the much larger Test 2.

Table 6.6: Automatic Evaluation Results - BLEU

	Test 1	**Test 2**
STS	81.61	77.14
ED	77.32	81.34

Table 6.7: Automatic Evaluation Results - METEOR

	Test 1	**Test 2**
STS	92.6	87.35
ED	91.5	84.55

As the automatic evaluation results are inconsistent depending on the chosen test set, we perform a full manual evaluation on the data in order to gain more insight into the usefulness of STS in MT retrieval. We present our retrieved matches to a native speaker of English, alongside the original sentence and the ED match. The annotator was asked to rank the sentences with the following labels:

0 If the two sentences are of the same quality.

1 If the ED retrieved match is a better match.

2 If the STS retrieved match is a better match.

The sentences are rated on informativeness. That is, they were rated on how closely they preserved the original meaning of the sentence. We evaluate 1,000 sentences this way. Table 6.8 shows the percentage of sentences that the annotator ranked as better for each category. We also group these sentences by ED range, in order to get

a better idea for when the STS system performs better than the ED matches. In the overwhelming majority of cases, we find that the annotator found no difference between the quality of the ED and the STS match. That is, cases where the annotator labelled "0" were still the overwhelming majority. In cases of high ED match, the ED system seems to perform slightly better than the STS system, with 3.16% of the ED sentences being the better match, as opposed to only 0.45% of the STS group. When we control for lower ED matches, the STS system starts to perform slightly better. In the 24-50 ED range, the annotator chose 16% of the STS sentences over the ED sentences and only 6.2% of the ED sentences as better. Usually, segments with such low ED matches are not considered useful to the posteditor.

Table 6.8: Manual Analysis - Percentage of sentences for which STS/ED retrieved the better match

Range	**STS Matches**	**ED Matches**
75-100	0.45%	3.16%
50-75	2.94%	2.94%
25-50	16%	6.2%

A closer look at the data shows that sentences in the 75-100% ED range would almost always retrieve identical sentences for both the ED and the STS match. In the handful of cases where the match is not identical, the sentence retrieved through ED is almost always the closer match. Example (1) shows such a case, where the manual evaluation denotes that the ED match is a better choice.

(1) a. **Source:** This Decision shall enter into force on the date of its publication in the Official Journal of the European Union .

b. **ED Match [95]**: This Decision shall enter into force on the day of its publication in the Official Journal of the European Union .

c. **STS Match [4.5]**: This Decision shall enter into force on the date of its adoption .

Between 76-100% ED, there are no cases where the manual evaluator chose the STS match over the ED match. The first such case happens at ED 75%, in Example (2). Here the STS match clearly is closer in meaning, even though the ED match is closer on the surface. In this case, the STS system manages to capture a better match for the user.

(2) a. **Source**: Detailed provisions for performance requirements for EDS are laid down in a separate Commission Decision .

b. **ED Match [75]**: Detailed provisions for a hand search are laid down in a separate Commission Decision .

c. **STS Match [4.29]**: The performance requirements for an EDD are laid down in Attachment 12-D of a separate Commission Decision .

Once exact or near exact matches become unavailable, however, the instances for which the manual evaluator chose the STS match become more numerous. In Examples (3) and (4), the manual evaluator chose the STS match over the ED match.

(3) a. **Source:** The list of experts and the subject of the tasks shall be published annually .

b. **ED Match [35]:** In the case of calves slaughtered before the age of 3 months , the retention period shall be 1 month .

c. **STS Match [3.7]:** The final accounts shall be published.

(4) a. **Source:** This was confirmed by the positive development of its economic situation observed during the period considered .

b. **ED Match [35]:** The table below shows the development of car production volumes in Europe in the period considered .

c. **STS Match [3.0]** The results of this rationalisation process within the Community industry was thus reflected in the productivity which was

rather stable during the period considered .

In examples (3) and (4), the ED match is very low (35). This score is too low to produce a usable match. We can see that in cases like this, where no higher ED match can be found, it would make more sense to use the STS match. This human evaluation remains limited, however, as TM matches are more often used for post-editing. Therefore, a stronger manual evaluation involving post-editing would be required to give us a deeper insight into the usefulness of these matches.

6.3.4 Summary

In this section, we presented our approach to using STS to retrieve translation memory segments. We compared our system's performance to that of word-based edit distance, and achieved comparable results. While our system did not outperform the baseline in cases where the edit distance matches were high (35% or higher), it did manage to perform better for segments where only low matches (under 35%) were found. Our system faces a number of limitations, due to the limited availability of suitable corpora and the time it takes to search large corpora for matching sentences. Furthermore, we trained the STS system on the SICK dataset. This training data was not in the same domain as the DGT-TM, which was used for testing. This out-of-domain training data affected the accuracy of the STS system. However, more suitable training data was not readily available.

6.4 Conclusion

In this chapter, we set out to find further applications for STS as a tool for evaluation, and aimed to answer research question RQ3.

[RQ3] Can we expand the applications of Semantic Textual Similarity further:

RQ3.1: in automatic evaluation of simplified text?

RQ3.2: in translation memory matching and retrieval?

We answered research question RQ3.1 through our submission to the QATS workshop, which called for participants to submit automatic evaluation metrics and quality estimation systems to evaluate the output of automatic text simplification systems. Our submission consisted of a machine learning system built on STS features. As these features only capture similarity, we added MTQE features and readability metrics to address fluency and simplicity. Our overall system came in third for "Meaning Preservation". However, the system did not perform as well for fluency (Grammaticality in the shared task) and simplicity. This shows that while STS features can adequately indicate whether or not a simplified sentence captures the meaning of the original, it only covers one aspect of text simplification. The other two aspects would require further evaluation.

In this chapter, we also addressed research question RQ3.2, this time with a study that compares the STS tool to Edit Distance metrics in TM matching and retrieval. Using the DGT–TM as our dataset, we extracted a set of testing segments and a set to use as our TM. For each segment in our test set, we selected the most semantically similar segment from the TM, and compared the result to the sentence retrieved by ED. We performed both a manual and automatic evaluation, and found that overall, STS is not as useful or as effective as basic ED. However, in some cases where no useful ED match can be found, STS was able to retrieve a match.

Chapter 7

Conclusions

This thesis presented our research into the applications of semantic textual similarity in the evaluation of several NLP tasks, and detailed our findings. Chapter 2 presented background information and the state of the art for the main topics covered in our research. This included a detailed look at the history and developement of semantic textual similarity and recognising textual entailment. Additionally, it provided a look at the state of the art in machine translation evaluation, from MT evaluation metrics to quality estimation methods. Chapter 3 presented several supervised machine learning approaches to determining semantic textual similarity that exploited existing language technology. Chapter 4 presented a novel approach to using semantic textual similarity for machine translation evaluation. Chapter 5 tested our system in a real-world setting through a user study that presented professional translators with MT suggestions and a traffic lights system that reflected MTQE information about the usefulness of these suggestions. Chapter 6 presented two more applications for STS, in the fields of automatic evaluation of text simplification, and in translation memory matching and retrieval.

7.1 Research Questions Revisited

At the beginning of this thesis, we posed the following research questions:

RQ1 Can semantic textual similarity help accurately predict the quality of MT output?

The question of whether semantic textual similarity can help accurately predict the quality of MT output is researched in this thesis. This question is mostly tackled in Chapter 4 with the outcome of our experiments that show that the introduction of semantic information into the evaluation process can indeed improve on the baseline. In this chapter we attempted to predict the quality of a MT sentence by comparing it to a semantically similar sentence that has been previously evaluated. We tested this approach on 3 different datasets, including a dataset of our own design. Our results showed a small improvement over the baseline when using the STS enhanced methods, with all three sets of experiments showing minor improvements to the baseline when augmented with STS features.

RQ2 To what extent does the use of quality estimation tools affect the efficiency of the translation workflow?

This question is answered in Chapter 5, where we describe a user study that investigates the integration of MTQE into the postediting workflow. We enlist the help of professional translators and ask them to postedit or translate sentences in a controlled environment. This is achieved by way of a traffic light system, using the PET post-editing tool. We present the user with three different categories of sentences: sentences to translate without a machine translation, sentences to post-edit without any MTQE suggestion and sentences to either post-edit or translate depending on the MTQE suggestion. Our findings show that not only does MTQE information improve the effi-

ciency of translators in terms of time and effort, but that good and accurate MTQE specifically cuts post-editing time and effort by up to 13%.

RQ3 Can we explore the applications of Semantic Textual Similarity further:

RQ3.1 in automatic evaluation of simplified text?

RQ3.2 in translation memory matching and retrieval?

We address both parts of the final research question in Chapter 6, which is itself split into two major sections. The first section deals with the question of automatic text simplification. We apply our STS tool to the problem of determining the quality of automatic text simplification. We look at three aspects of quality: simplicity, meaning preservation (which relates to adequacy) and grammaticality (which relates to fluency). As our STS tool can only determine meaning preservation, we augment our tool with simplicity features based on readability indices, and with MTQE features to determine grammaticality. The system performs well on meaning preservation, showing that semantic textual similarity can be used to determine at least one aspect of quality. The second part of the chapter investigates the use of STS in translation memory matching and retrieval, comparing it to a basic edit distance tool. We use a translation memory (DGT-TM) to find the best matches using both edit distance and the STS tool. We then choose the better match, using both automatic and manual evaluation. Our evaluation shows that while edit distance generally finds a better match, STS can be useful in finding a match in cases where no good match (ED < 35) is available. However, the extent to which these sentences are useful to a post-editor would require further manual evaluation.

7.2 Contributions

The main contributions of this thesis are as follows:

- Our literature survey covers both the state of the art in semantic textual similarity and machine translation evaluation. It reviews the different methods and contributions of the most prominent researchers in both fields.
- We add to the STS field by proposing our own machine learning approach to determining semantic textual similarity
- We add to the MTQE field in two ways:
  - By proposing a novel STS-enhanced MTQE method and extensively testing its performance.
  - By testing the impact of MTQE on post-editing efficiency in a real-world setting
- We add to the field of automatic text simplification by proposing a method with which to evaluate meaning preservation after simplification.

7.3 Future Work

While this thesis adequately answers the questions it posed in the introductory chapter, there are several open questions and issues that may be investigated in the future. Some of these questions are outlined below:

- Our STS tool is specifically designed to calculate the similarity between pairs of sentences written in English. One possible avenue of future work would be to modify the tool for other languages. Research into language-independent

STS tools is still fairly limited, although its uses, as demonstrated in this thesis, are quite vast.

- Furthering our experiments in Chapter 4, another interesting question is whether the STS score carries over post-translation, and how much translation quality affects the degree to which the STS score carries over. Such experiments would require an annotated dataset with manually provided STS scores both before and after machine translation.

- One important avenue of future work would be to test whether the results in Chapter 5 can be replicated for other language pairs and domains. Our experiments are limited to the English-Spanish language pair. Similar findings in other language pairs experiments would demonstrate the need for accurate and reliable MTQE, as well as the need to integrate it in professional translation workflows to improve post-editing efficiency.

- Furthermore, the user study described in Chapter 5 only includes 4 professional translators and 260 segments to translate. While expanding the scope of translators and segments would be quite challenging and expensive, it would provide a more robust and conclusive picture of the effect of MTQE.

Bibliography

Agirre, E., Banea, C., Cardie, C., Cer, D., Diab, M., Gonzalez-Agirre, A., Guo, W., Lopez-Gazpio, I., Maritxalar, M., Mihalcea, R., et al. (2015). SemEval–2015 task 2: Semantic textual similarity, English, Spanish and pilot on interpretability. In *Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)*, pages 252–263.

Agirre, E., Banea, C., Cer, D., Diab, M., Gonzalez-Agirre, A., Mihalcea, R., Rigau, G., and Wiebe, J. (2016). SemEval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 497–511.

Agirre, E., Cer, D., Diab, M., Gonzalez-Agirre, A., and Guo, W. (2013). * SEM 2013 shared task: Semantic textual similarity. In *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, volume 1, pages 32–43.

Agirre, E., Diab, M., Cer, D., and Gonzalez-Agirre, A. (2012). SemEval-2012 task 6: A pilot on semantic textual similarity. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, pages 385–393. Association for Computational Linguistics.

Aliguliyev, R. M. (2009). A new sentence similarity measure and sentence based extractive technique for automatic text summarization. *Expert Systems with Applications*, 36(4):7764–7772.

Aziz, W., de Souza, S. C. M., and Specia, L. (2012). PET: A Tool for Post-Editing and Assessing Machine Translation. In *Eighth International Conference on Language Resources and Evaluation*, pages 3982 – 3987, Istanbul, Turkey.

Baker, M. (2011). *In Other Words: A Coursebook on Translation*. Routledge, United Kingdom, 2nd edition.

Banerjee, S. and Lavie, A. (2005). METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, volume 29, pages 65–72.

Bär, D., Biemann, C., Gurevych, I., and Zesch, T. (2012). UKP: Computing semantic textual similarity by combining multiple content similarity measures. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, pages 435–440. Association for Computational Linguistics.

Bar-Haim, R., Dagan, I., Dolan, B., Ferro, L., Giampiccolo, D., Magnini, B., and Szpektor, I. (2006). The second PASCAL recognising textual entailment challenge. In *Proceedings of the second PASCAL challenges workshop on recognising textual entailment*, volume 6, pages 6–4. Venice.

Béchara, H., Costa, H., Taslimipoor, S., Gupta, R., Orăsan, C., Pastor, G. C., and Mitkov, R. (2015). MiniExperts: An SVM approach for measuring semantic textual similarity. In *Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)*, pages 96–101.

Béchara, H., Gupta, R., Tan, L., Orasan, C., Mitkov, R., and van Genabith, J. (2016). Wolvesaar at SemEval-2016 task 1: Replicating the success of monolingual word alignment and neural embeddings for semantic textual similarity. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 634–639.

Bentivogli, L., Dagan, I., Dang, H. T., Giampiccolo, D., and Magnini, B. (2009). The Fifth PASCAL Recognizing Textual Entailment Challenge. In *Proceedings of the 2009 Text Analysis Conference (TAC'09)*.

Bentivogli, L., Federico, M., Moretti, G., and Paul, M. (2011). Getting expert quality from the crowd for machine translation evaluation. *Proceedings of the MT Summmit*, 13:521–528.

Best, C., van der Goot, E., Blackler, K., Garcia, T., and Horby, D. (2005). Europe media monitor. *Technical Report EUR221 73 EN, European Commission*.

Bhatia, P. K., Mathur, T., and Gupta, T. (2013). Survey paper on information retrieval algorithms and personalized information retrieval concept. *International Journal of Computer Applications*, 66(6).

Biçici, E. (2013). Referential translation machines for quality estimation. In *Proceedings of the eighth workshop on statistical machine translation*, pages 343–351.

Biçici, E. and Van Genabith, J. (2013). CNGL-CORE: Referential translation machines for measuring semantic similarity. In *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, volume 1, pages 234–240.

Biçici, E. and Yuret, D. (2011). Instance selection for machine translation using feature decay algorithms. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 272–283. Association for Computational Linguistics.

Biran, O., Brody, S., and Elhadad, N. (2011). Putting it simply: a context-aware approach to lexical simplification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 496–501. Association for Computational Linguistics.

Bjerva, J., Bos, J., Van der Goot, R., and Nissim, M. (2014). The meaning factory: Formal semantics for recognizing textual entailment and determining semantic similarity. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 642–646.

Bjornsson, C. (1968). Lasbarhet. *Liber, Stockholm*.

Blatz, J., Fitzgerald, E., Foster, G., Gandrabur, S., Goutte, C., Kulesza, C., Sanchis, A., and Ueffing, N. (2004). Confidence estimation for machine translation. In *Proceedings of the 20th International Conference on Computational Linguistics (CoLing-2004)*, pages 315–321.

Bojar, O. (2011). Analyzing error types in English-Czech machine translation. *The Prague Bulletin of Mathematical Linguistics*, 95:63–76.

Bojar, O., Buck, C., Callison-Burch, C., Federmann, C., Haddow, B., Koehn, P., Monz, C., Post, M., Soricut, R., and Specia, L. (2013). Findings of the 2013 Workshop on Statistical Machine Translation. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 1–44, Sofia, Bulgaria. Association for Computational Linguistics.

Bojar, O., Buck, C., Federmann, C., Haddow, B., Koehn, P., Leveling, J., Monz, C., Pecina, P., Post, M., Saint-Amand, H., et al. (2014). Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 12–58. Association for Computational Linguistics Baltimore, MD, USA.

Bojar, O., Chatterjee, R., Federmann, C., Graham, Y., Haddow, B., Huang, S., Huck, M., Koehn, P., Liu, Q., Logacheva, V., et al. (2017). Findings of the

2017 conference on machine translation (wmt17). In *Proceedings of the Second Conference on Machine Translation*, pages 169–214.

Bojar, O., Chatterjee, R., Federmann, C., Graham, Y., Haddow, B., Huck, M., Yepes, A. J., Koehn, P., Logacheva, V., Monz, C., et al. (2016). Findings of the 2016 conference on machine translation. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, volume 2, pages 131–198.

Bojar, O., Chatterjee, R., Federmann, C., Haddow, B., Huck, M., Hokamp, C., Koehn, P., Logacheva, V., Monz, C., Negri, M., Post, M., Scarton, C., Specia, L., and Turchi, M. (2015). Findings of the 2015 workshop on statistical machine translation. In *Tenth Workshop on Statistical Machine Translation*, pages 1–46, Lisboa, Portugal.

Bos, J. and Markert, K. (2006). When logical inference helps determining textual entailment (and when it doesn't). In *Proceedings of the Second PASCAL RTE Challenge*, page 26.

Bowman, S. R., Angeli, G., Potts, C., and Manning, C. D. (2015). A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*.

Brown, P., Pietra, J., Pietra, S. D., Jelinek, F., Mercer, R., , and Roossin, P. (1990). A Statistical Approach to Machine Translation. In *Computational Linguistics*, pages 16:79–85.

Brown, P. F., Della-Pietra, S. A., Della-Pietra, V. J., and Mercer, R. L. (1993). The mathematics of statistical machine translation. *Computational Linguistics*, 19(2):263–313.

Brychcín, T. and Svoboda, L. (2016). UWB at SemEval-2016 Task 1: Semantic textual similarity using lexical, syntactic, and semantic information. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 588–594. Association for Computational Linguistics.

Callison-Burch, C., Fordyce, C., Koehn, P., Monz, C., and Schroeder, J. (2007). (Meta-) evaluation of machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 136–158. Association for Computational Linguistics.

Callison-Burch, C., Fordyce, C., Koehn, P., Monz, C., and Schroeder, J. (2008). Further Meta-Evaluation of Machine Translation. In *Proceedings of the Third Workshop on Statistical Machine Translation (WMT)*, pages 70–106.

Callison-Burch, C., Koehn, P., Monz, C., Peterson, K., Przybocki, M., and Zaidan, O. F. (2010). Findings of the 2010 joint workshop on statistical machine translation and metrics for machine translation. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 17–53. Association for Computational Linguistics.

Callison-Burch, C., Koehn, P., Monz, C., Post, M., Soricut, R., and Specia, L., editors (2012). *Proceedings of the Seventh Workshop on Statistical Machine Translation*. Association for Computational Linguistics, Montréal, Canada.

Callison-Burch, C., Koehn, P., Monz, C., and Schroeder, J. (2009). Findings of the 2009 workshop on statistical machine translation. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 1–28. Association for Computational Linguistics.

Callison-Burch, C., Koehn, P., Monz, C., and Zaidan, O. F. (2011). Findings of the 2011 workshop on statistical machine translation. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 22–64. Association for Computational Linguistics.

Castillo, J. and Estrella, P. (2012a). Semantic textual similarity for mt evaluation. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 52–58. Association for Computational Linguistics.

Castillo, J. and Estrella, P. (2012b). Semantic Textual Similarity for MT Evaluation. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, WMT '12, pages 52–58, Stroudsburg, PA, USA. Association for Computational Linguistics.

Cer, D., Diab, M., Agirre, E., Lopez-Gazpio, I., and Specia, L. (2017). Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation. *arXiv preprint arXiv:1708.00055*.

Chang, C.-C. and Lin, C.-J. (2011a). Libsvm: A library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):27.

Chang, C.-C. and Lin, C.-J. (2011b). Libsvm: A library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):27.

Chen, D. L. and Dolan, W. B. (2011). Collecting highly parallel data for paraphrase evaluation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 190–200. Association for Computational Linguistics.

Clinchant, S., Goutte, C., and Gaussier, E. (2006). Lexical entailment for information retrieval. In *European Conference on Information Retrieval*, pages 217–228. Springer.

Clough, P. and Stevenson, M. (2011). Developing a corpus of plagiarised short answers. *Language Resources and Evaluation*, 45(1):5–24.

Coleman, E. (1971). Developing a technology of written instruction: some determiners of the complexity of prose. *Verbal Learning Research Comprehension and the Technology of Written Instruction: Teachers College Press*, pages 155–204.

Coster, W. and Kauchak, D. (2011). Learning to simplify sentences using Wikipedia. In *Proceedings of the workshop on monolingual text-to-text generation*, pages 1–9. Association for Computational Linguistics.

Cruse, D., Cruse, D., Cruse, D., Cruse, D., Anderson, S., Bresnan, J., Comrie, B., Dressler, W., and Ewen, C. (1986). *Lexical Semantics*. Cambridge Textbooks in Linguistics. Cambridge University Press.

Dagan, I., Dolan, B., Magnini, B., and Roth, D. (2010). Recognizing textual entailment: Rational, evaluation and approaches – Erratum. *Natural Language Engineering*, 16(1):105–105.

Dagan, I., Glickman, O., and Magnini, B. (2005). The PASCAL recognising textual entailment challenge. In *Proceedings of the First International Conference on Machine Learning Challenges: Evaluating Predictive Uncertainty Visual Object Classification, and Recognizing Textual Entailment*, MLCW'05, pages 177–190, Berlin, Heidelberg. Springer-Verlag.

De Marneffe, M.-C., Rafferty, A. N., and Manning, C. D. (2008). Finding contradictions in text. *Proceedings of ACL-08: HLT*, pages 1039–1047.

de Souza, J. G. C., González-Rubio, J., Buck, C., Turchi, M., and Negri, M. (2014). FBK-UPV-UEdin participation in the WMT14 quality estimation shared-task. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 322–328, Baltimore, Maryland, USA. Association for Computational Linguistics.

Doddington, G. (2002). Automatic Evaluation of Machine Translation Quality Using N-gram Co-occurrence Statistics. In *Proceedings of the Second International Conference on Human Language Technology Research*, HLT '02, pages 138–145, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Dolan, B., Quirk, C., and Brockett, C. (2004). Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In *Proceedings of the 20th international conference on Computational Linguistics*, page 350. Association for Computational Linguistics.

Dzikovska, M. O., Bental, D., Moore, J. D., Steinhauser, N. B., Campbell, G. E., Farrow, E., and Callaway, C. B. (2010). Intelligent tutoring with natural language

support in the Beetle II system. In *European Conference on Technology Enhanced Learning*, pages 620–625. Springer.

Evans, R. and Orăsan, C. (2019). Identifying signs of syntactic complexity for rule-based sentence simplification. *Natural Language Engineering*, 25(1):69–119.

Fajardo, I., Ávila, V., Ferrer, A., Tavares, G., Gómez, M., and Hernández, A. (2014). Easy-to-read texts for students with intellectual disability: Linguistic factors affecting comprehension. *Journal of Applied Research in Intellectual Disabilities*, 27(3):212–225.

Federico, M., Bertoldi, N., Cettolo, M., Negri, M., Turchi, M., Trombetti, M., Cattelan, A., Farina, A., Lupinetti, D., Martines, A., et al. (2014). The matecat tool. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: System Demonstrations*, pages 129–132.

Ferrés, D., Marimon, M., and Saggion, H. (2015). A web-based text simplification system for english. *Procesamiento del Lenguaje Natural*, (55).

Flesch, R. (1948). A new readability yardstick. *Journal of applied psychology*, 32(3):221.

Gabrilovich, E. and Markovitch, S. (2007). Computing semantic relatedness using Wikipedia-based explicit semantic analysis. In *IJcAI*, volume 7, pages 1606–1611.

Gandrabur, S. and Foster, G. (2003). Confidence estimation for translation prediction. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4*, CONLL '03, pages 95–102, Stroudsburg, PA, USA. Association for Computational Linguistics.

Ganitkevitch, J., Van Durme, B., and Callison-Burch, C. (2013). Ppdb: The paraphrase database. In *HLT-NAACL*, pages 758–764.

Giampiccolo, D., Dang, H. T., Magnini, B., Dagan, I., Cabrio, E., and Dolan, B. (2008). The fourth Pascal recognizing textual entailment challenge. In *Pro-*

ceedings of the Text Analysis Conference 2008 Workshop on Textual Entailment (TAC'2008).

Giampiccolo, D., Magnini, B., Dagan, I., and Dolan, B. (2007). The third pascal recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL workshop on textual entailment and paraphrasing*, pages 1–9. Association for Computational Linguistics.

Giménez, J. and Màrquez, L. (2007). Linguistic features for automatic evaluation of heterogenous MT systems. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 256–264. Association for Computational Linguistics.

Glavaš, G. and Štajner, S. (2015). Simplifying lexical simplification: do we need simplified corpora? In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, volume 2, pages 63–68.

Goto, S., Lin, D., and Ishida, T. (2014). Crowdsourcing for evaluating machine translation quality. In *LREC*, volume 2014, pages 3456–346.

Graham, Y., Mathur, N., and Baldwin, T. (2015). Accurate evaluation of segment-level machine translation metrics. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics Human Language Technologies*, pages 1183–1191.

Gunning, R. (1952). The technique of clear writing. *McGraw Hill, New York*.

Gupta, R., Béchara, H., El Maarouf, I., and Orăsan, C. (2014). Uow: Nlp techniques developed at the university of wolverhampton for semantic similarity and textual entailment. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 785–789.

Gupta, R. and Orăsan, C. (2014). Incorporating paraphrasing in translation memory matching and retrieval. In *Proceedings of the 17th Annual Conference of European*

Association for Machine Translation, Croatian Language Technologies Society, Dubrovnik, Croatia, pages 3–10.

Gupta, R., Orăsan, C., and van Genabith, J. (2015a). ReVal: a simple and effective machine translation evaluation metric based on recurrent neural networks. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1066–1072.

Gupta, R., Orăsan, C., Zampieri, M., Vela, M., and Van Genabith, J. (2015b). Can translation memories afford not to use paraphrasing? In *Proceedings of the 18th Annual Conference of the European Association for Machine Translation*.

Gupta, R., Orăsan, C., Zampieri, M., Vela, M., van Genabith, J., and Mitkov, R. (2016). Improving translation memory matching and retrieval using paraphrases. *Machine Translation*, 30(1-2):19–40.

Han, A. L.-F. and Wong, D. F. (2016). Machine translation evaluation: A survey. *arXiv preprint arXiv:1605.04515*.

Han, L., Kashyap, A., Finin, T., Mayfield, J., and Weese, J. (2013). UMBC EBIQUITY-CORE: Semantic textual similarity systems. *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, page 44.

Hänig, C., Remus, R., and De La Puente, X. (2015). Exb themis: Extensive feature extraction from word alignments for semantic textual similarity. In *Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)*, pages 264–268.

Hanks, P. (2013). Lexical Analysis: Norms and Exploitations.

Harabagiu, S. and Hickl, A. (2006). Methods for using textual entailment in open-domain question answering. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for*

Computational Linguistics, pages 905–912. Association for Computational Linguistics.

He, Y., Ma, Y., van Genabith, J., and Way, A. (2010). Bridging SMT and TM with Translation Recommendation. In *Proceedings of the 28th Annual Meeting of the Association for Computational Linguistics*, pages 622–630.

Hickl, A. and Bensley, J. (2007). A discourse commitment-based framework for recognizing textual entailment. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 171–176. Association for Computational Linguistics.

Hickl, A., Williams, J., Bensley, J., Roberts, K., Rink, B., and Shi, Y. (2006). Recognizing textual entailment with LCC's GROUNDHOG system. In *Proceedings of the Second PASCAL Challenges Workshop*, volume 18.

Hodász, G. and Pohl, G. (2005). MetaMorpho TM: a linguistically enriched translation memory. In *In International Workshop, Modern Approaches in Translation Technologies*.

Horn, C., Manduca, C., and Kauchak, D. (2014). Learning a lexical simplifier using Wikipedia. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 458–463.

Ive, J., Blain, F., and Specia, L. (2018). DeepQuest: a framework for neural-based quality estimation. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3146–3157.

Jaccard, P. (1901). Étude comparative de la distribution florale dans une portion des alpes et des jura. *Bulletin del la Société Vaudoise des Sciences Naturelles*, 37:547–579.

Jimenez, S., Becerra, C., and Gelbukh, A. (2012). Soft cardinality+ ML: learning adaptive similarity functions for cross-lingual textual entailment. In *Proceedings*

of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation, pages 684–688. Association for Computational Linguistics.

Jordanova, V., Evans, E., and Cerga-Pashoja, A. (2013). First deliverable - benchmark report (result of piloting task). In *Technical Report D7.2, Central and Northwest London NHS Foundation Trust, London, UK.*

Kaljahi, R., Foster, J., and Roturier, J. (2014). Syntax and semantics in quality estimation of machine translation. *Syntax, Semantics and Structure in Statistical Translation*, page 67.

Kilgarriff, A. (2001). Comparing Corpora. *Int. Journal of Corpus Linguistics*, 6(1):97–133.

Kim, H., Lee, J.-H., and Na, S.-H. (2017). Predictor-estimator using multilevel task learning with stack propagation for neural quality estimation. In *Proceedings of the Second Conference on Machine Translation*, pages 562–568.

Koehn, P. (2005). Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. (2007). Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the Association for Computational Linguistics (ACL)*, pages 177–180.

Koehn, P. and Monz, C. (2006). Manual and automatic evaluation of machine translation between European languages. In *Proceedings of the Workshop on Statistical Machine Translation*, pages 102–121. Association for Computational Linguistics.

Koehn, P., Och, F., and Marcu, D. (2003). Statistical Phrase-Based Translation. In *Proceedings of the Human Language Technology Conference and the North American Chapter of the Association for Computational Linguistics (HLT/NAACL)*, pages 48–54.

Kulesza, A. and Shieber, S. M. (2004). A learning approach to improving sentence-level MT evaluation. In *In Proceedings of the 10th International Conference on Theoretical and Methodological Issues in Machine Translation (TMI)*.

Langlais, P., Foster, G., and Lapalme, G. (2000). TransType: a computer-aided translation typing system. In *Proceedings of the 2000 NAACL-ANLP Workshop on Embedded machine translation systems-Volume 5*, pages 46–51. Association for Computational Linguistics.

Lee, Y.-J., Yeh, Y.-R., and Pao, H.-K. (2010). An introduction to support vector machines. *National Taiwan University of Science and Technology*.

Lin, C.-Y. and Och, F. J. (2004). Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, page 605. Association for Computational Linguistics.

Lin, D. (1998). An Information-Theoretic Definition of Similarity. In *15th Int. Conf. on Machine Learning*, ICML'98, pages 296–304, San Francisco, CA, USA. Morgan Kaufmann.

Lo, C.-K. and Wu, D. (2011). MEANT: An inexpensive, high-accuracy, semi-automatic metric for evaluating translation utility via semantic frames. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 220–229. Association for Computational Linguistics.

Macklovitch, E. and Russell, G. (2000). What's been forgotten in translation memory. In *Proceedings of the 4th Conference of the Association for Machine Trans-*

lation in the Americas on Envisioning Machine Translation in the Information Future, AMTA '00, pages 137–146, London, UK, UK. Springer-Verlag.

Manning, C., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S., and McClosky, D. (2014). The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60.

Marelli, M., Bentivogli, L., Baroni, M., Bernardi, R., Menini, S., and Zamparelli, R. (2014a). SemEval-2014 Task 1: Evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment. In *8^{th} Int. Workshop on Semantic Evaluation*, SemEval-2014, Dublin, Ireland.

Marelli, M., Menini, S., Baroni, M., Bentivogli, L., Bernardi, R., and Zamparelli, R. (2014b). A SICK cure for the evaluation of compositional distributional semantic models. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 216–223, Reykjavik, Iceland. European Language Resources Association (ELRA).

Mc Laughlin, G. H. (1969). Smog grading-a new readability formula. *Journal of reading*, 12(8):639–646.

Mihalcea, R., Corley, C., and Strapparava, C. (2006). Corpus-based and knowledge-based measures of text semantic similarity. In *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 1*, AAAI'06, pages 775–780. AAAI Press.

Mirkin, S., Specia, L., Cancedda, N., Dagan, I., Dymetman, M., and Szpektor, I. (2009). Source-language entailment modeling for translating unknown terms. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 791–799. Association for Computational Linguistics.

Mohler, M., Bunescu, R., and Mihalcea, R. (2011). Learning to grade short answer questions using semantic similarity measures and dependency graph alignments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 752–762. Association for Computational Linguistics.

Moorkens, J. and O'Brien, S. (2017). *Human Issues in Translation Technology: The IATIS Yearbook*, chapter Assessing User Interface Needs of Post-Editors of Machine Translation, pages 109–130. Routledge, Oxford, UK.

Moorkens, J., O'Brien, S., A.L. da Silva, I., B. de Lima Fonseca, N., and Alves, F. (2015). Correlations of perceived post-editing effort with measurements of actual effort. *Machine Translation*, 29(3–4):267–284.

Moorkens, J. and Way, A. (2016). Comparing Translator Acceptability of TM and SMT outputs. *Baltic Journal of Modern Computing*, 4(2):141–151.

Navigli, R. and Ponzetto, S. P. (2012). Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250.

Och, F. (2003). Minimum Error Rate Training in Statistical Machine Translation. In *Proceedings of the Association for Computational Linguistics (ACL)*, pages 160–167.

Och, F. and Ney, H. (2003). A Systematic Comparison of Various Statistical Alignment Models. In *Proceedings of the Association for Computer Linguistics (ACL)*, pages 29(1):19–51.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Parra Escartín, C. and Arcedillo, M. (2015a). Living on the edge: productivity gain thresholds in machine translation evaluation metrics. In *Proceedings of the Fourth Workshop on Post-editing Technology and Practice*, pages 46–56, Miami, Florida (USA). Association for Machine Translation in the Americas (AMTA).

Parra Escartín, C. and Arcedillo, M. (2015b). Machine translation evaluation made fuzzier: A study on post-editing productivity and evaluation metrics in commercial settings. In *Proceedings of the MT Summit XV*, Miami (Florida). International Association for Machine Translation (IAMT).

Pilehvar, M. T., Jurgens, D., and Navigli, R. (2013). Align, Disambiguate and Walk: A Unified Approach for Measuring Semantic Similarity. In *51st Annual Meeting of the Association for Computational Linguistics - Volume 1*, pages 1341–1351, Sofia, Bulgaria. ACL.

Planas, E. and Furuse, O. (2000). Multi-level similar segment matching algorithm for translation memories and example-based machine translation. In *Proceedings of the 18th conference on Computational linguistics-Volume 2*, pages 621–627. Association for Computational Linguistics.

Quirk, C. B. (2004). Training a sentence-level machine translation confidence metric. *Proceedings of the 4th Conference on Language Resources and Evaluation*, 4:825–828.

Ramaprabha, J., Das, S., and Mukerjee, P. (2018). Survey on sentence similarity evaluation using deep learning. In *Journal of Physics: Conference Series*, volume 1000, page 012070. IOP Publishing.

Rashtchian, C., Young, P., Hodosh, M., and Hockenmaier, J. (2010). Collecting image annotations using amazon's mechanical turk. pages 139–147. Association for Computational Linguistics.

Rello, L., Baeza-Yates, R., Dempere-Marco, L., and Saggion, H. (2013). Frequent words improve readability and short words improve understandability for people

with dyslexia. In *IFIP Conference on Human-Computer Interaction*, pages 203–219. Springer.

Rennes, E. and Jönsson, A. (2015). A tool for automatic simplification of swedish texts. In *Proceedings of the 20th Nordic Conference of Computational Linguistics (NODALIDA 2015)*, pages 317–320.

Rubino, R., Souza, J. G. C., Foster, J., and Specia, L. (2013). Topic models for translation quality estimation for gisting purposes. In *Machine Translation Summit XIV, pages 295–302.*

Rychalska, B., Pakulska, K., Chodorowska, K., Walczak, W., and Andruszkiewicz, P. (2016). Samsung Poland NLP Team at SemEval-2016 Task 1: Necessity for diversity; combining recursive autoencoders, WordNet and ensemble methods to measure semantic similarity. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 602–608.

Sammons, M., Vydiswaran, V., and Roth, D. (2011). Recognizing textual entailment. *Multilingual Natural Language Applications: From Theory to Practice. Prentice Hall, Jun.*

Šarić, F., Glavaš, G., Karan, M., Šnajder, J., and Bašić, B. D. (2012). Takelab: Systems for measuring semantic text similarity. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, pages 441–448. Association for Computational Linguistics.

Schneider, D., Zampieri, M., and van Genabith, J. (2018). Translation Memories and the Translator: A report on a user survey. *Babel*, 64(5-6):734–762.

Senter, R. and Smith, E. A. (1967). Automated readability index. Technical report, Cincinnati University, OH.

Shardlow, M. (2014). A survey of automated text simplification. *International Journal of Advanced Computer Science and Applications*, 4(1):58–70.

Siddharthan, A. and Mandya, A. (2014). Hybrid text simplification using synchronous dependency grammars with hand-written and automatically harvested rules. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 722–731.

Snover, M., Dorr, B., Schwartz, R., Micciulla, L., and J., M. (2006). A Study of Translation Edit Rate with Targeted Human Annotation. In *Proceedings of Association for Machine Translation in the Americas (AMTA)*, pages 223–231.

Soricut, R., Bach, N., and Wang, Z. (2012). The SDL language weaver systems in the WMT12 quality estimation shared task. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 145–151. Association for Computational Linguistics.

Specia, L. (2010). Translating from complex to simplified sentences. In *International Conference on Computational Processing of the Portuguese Language*, pages 30–39. Springer.

Specia, L. (2011). Exploiting objective annotations for measuring translation post-editing effort. In *Proceedings of the 15th Conference of the European Association for Machine Translation*, pages 73–80.

Specia, L., Hajlaoui, N., Hallett, C., and Aziz, W. (2011). Predicting machine translation adequacy. In *Machine Translation Summit XIII*, pages 513–520, Xiamen, China.

Specia, L., Paetzold, G., and Scarton, C. (2015). Multi-level Translation Quality Prediction with QuEst++. In *Proceedings of ACL-IJCNLP 2015 System Demonstrations*, pages 115–120, Beijing, China. Association for Computational Linguistics and The Asian Federation of Natural Language Processing.

Specia, L., Raj, D., and Turchi, M. (2010). Machine Translation Evaluation versus Quality Estimation. In *Machine Translation Volume 24, Issue 1*, pages 39–50.

Specia, L., Saunders, C., Turchi, M., Wang, Z., and Shawe-Taylor, J. (2009a). Improving the confidence of machine translation quality estimates. *Proceedings of the Twelfth Machine Translation Summit (MT Summit XII)*, pages 136–143.

Specia, L., Shah, K., Guilherme, J., de Souza, C., and Cohn, T. (2013). QuEst - A translation quality estimation framework. In *Proceedings of the Association for Computational Linguistics (ACL), Demonstrations*.

Specia, L., Turchi, M., Cancedda, N., Dymetman, M., and Cristianini, N. (2009b). Estimating the Sentence-Level Quality of Machine Translation Systems. In *13th Annual Meeting of the European Association for Machine Translation (EAMT-2009)*, pages 28–35.

Štajner, S., Béchara, H., and Saggion, H. (2015). A deeper exploration of the standard pb-smt approach to text simplification and its evaluation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, volume 2, pages 823–828.

Štajner, S., Mitkov, R., and Saggion, H. (2014). One step closer to automatic evaluation of text simplification systems. In *Proceedings of the 3rd workshop on predicting and improving text readability for target reader populations (PITR)@ EACL*, pages 1–10.

Štajner, S., Popovic, M., Specia, L., and Fishel, M. (2016). Shared task on quality assessment for text classification. In *In Proceedings of the LREC Workshop on Quality Assessment for Text Simplification (QATS)*.

Steinberger, R., Eisele, A., Klocek, S., Pilos, S., and Schlüter, P. (2013). DGT-TM: a freely available translation memory in 22 languages. *arXiv preprint arXiv:1309.5226*.

Steinberger, R., Pouliquen, B., Widiger, A., Ignat, C., Erjavec, T., Tufis, D., and Varga, D. (2006). The JRC-Acquis: a multilingual aligned parallel corpus with 20+ languages. *arXiv preprint cs/0609058*.

Stolcke, A. (2002). SRILM - an Extensible Language Modeling Toolkit. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, pages 901–904.

Su, K.-Y., Wu, M.-W., and Chang, J.-S. (1992). A new quantitative quality measure for machine translation systems. In *Proceedings of the 14th conference on Computational linguistics-Volume 2*, pages 433–439. Association for Computational Linguistics.

Sultan, M. A., Bethard, S., and Sumner, T. (2015). DLS@CU: sentence similarity from word alignment and semantic vector composition. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 148–153.

Teixeira, C. and O'Brien, S. (2017). The Impact of MT Quality Estimation on Post-Editing Effort. In *Proceedings of MT Summit XVI, Vol. 2: Users and Translators Track*, pages 211–233.

Tian, J., Zhou, Z., Lan, M., and Wu, Y. (2017). ECNU at SemEval-2017 Task 1: leverage kernel-based traditional nlp features and neural networks to build a universal model for multilingual and cross-lingual semantic textual similarity. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 191–197.

Turchi, M., Negri, M., and Federico, M. (2015). MT Quality Estimation for Computer-assisted Translation: Does it Really Help? In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Short Papers)*, pages 530–535, Beijing, China. Association for Computational Linguistics.

Turney, P. D. (2001). Mining the Web for Synonyms: PMI-IR Versus LSA on TOEFL. In *12th European Conf. on Machine Learning*, EMCL'01, pages 491–502, London, UK. Springer.

Ueffing, N., Macherey, K., and Ney, H. (2003). Confidence measures for Statistical Machine Translation. In *Proceedings of MT Summit IX, New Orleans*.

Vapnik, V. (2013). *The nature of statistical learning theory*. Springer science & business media.

Vilar, D., Matusov, E., Hasan, S., Zens, R., and Ney, H. (2005). Statistical machine translation of european parliamentary speeches. In *Proceedings of MT Summit X*, pages 259–266.

Vilar, D., Xu, J., d'Haro, L. F., and Ney, H. (2006). Error analysis of statistical machine translation output. In *Proceedings of LREC*, pages 697–702.

Woodsend, K. and Lapata, M. (2011). Learning to simplify sentences with quasi-synchronous grammar and integer programming. In *Proceedings of the conference on empirical methods in natural language processing*, pages 409–420. Association for Computational Linguistics.

Wu, H., Huang, H., Jian, P., Guo, Y., and Su, C. (2017). BIT at SemEval-2017 Task 1: using semantic information space to evaluate semantic textual similarity. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 77–84.

Wu, S., Zhu, D., Carterette, B., and Liu, H. (2013). Mayoclinicnlp–core: Semantic representations for textual similarity. *Atlanta, Georgia, USA*, page 148.

Yaneva, V. and Evans, R. (2015). Six good predictors of autistic text comprehension. In *Proceedings of the International Conference Recent Advances in Natural Language Processing*, pages 697–706.

Zaidan, O. F. and Callison-Burch, C. (2011). Crowdsourcing translation: Professional quality from non-professionals. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 1220–1229. Association for Computational Linguistics.

Zaretskaya, A., Pastor, G. C., and Seghiri, M. (2015). Translators' Requirements for Translation Technologies: A user survey. *New Horizons in Translation and Interpreting Studies*, pages 133–134.

Zhao, J., Zhu, T., and Lan, M. (2014). ECNU: One stone two birds: Ensemble of heterogenous measures for semantic relatedness and textual entailment. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 271–277.