eman ta zabal zazu

Universidad del País Vasco

Euskal Herriko Unibertsitatea

To post-edit or to translate… That is the question

A case study of a recommender system for Quality Estimation of Machine Translation based on linguistic features

Author: Ona de Gibert Bonet

Advisor: Nora Aranberri

hap/lap

Hizkuntzaren Azterketa eta Prozesamendua

Language Analysis and Processing

June 2018

Departments: Computer Systems and Languages, Computational Architectures and Technologies, Computational Science and Artificial Intelligence, Basque Language and Communication, Communications Engineer.

Laburpena

Itzulpen automatikoko sistema bat produkzio-katean sartzeak ez du bere horretan erabilera eraginkor bat bermatzen. Beharrezkoa da jakitea noiz den probetxugarria itzulpen automatikoa editatzea eta noiz eskuz itzultzea. Horretarako ezinbestekoa da itzulpen automatikoaren kalitatea aurreikusteko gai izatea. Lan honek ikertzen du itzulpen automatikoaren kalitatearen estimazioa sistema zehatz batentzat eta domeinu zehatz baterako, gomendio sistema bat garatuz gaztelaniatik ingelesera itzultzerakoan erabiltzeko. Lanean aztertzen da nola lagundu dezaketen ezaugarri linguistikoek kalitatearen estimazioan, ohikoak diren azaleko ezaugarriekin alderatuta. Datuak itzultzaile profesionalen postedizio lanetik bildu dira eta ezaugarri linguistikoak eskuz etiketatu. Lehenengo, esaldi bat posteditatzea edo itzultzea gomendatzen duten sailkapen ereduak eraiki dira. Bigarrenik, erregresio ereduak entrenatu dira hiru kalitate adierazle aurreikusteko: kalitatea, denbora eta HTER. Esperimentuek emaitza adierazgarriak erakusten dituzten arren, orokorrean erabilitako ezaugarriek ez dute behar bezala bereizten edizio mota komenigarriena zein den, eta beraz, gomendio sistemaren doitasuna ez da produkzioan ezartzeko nahikoa. Emaitzak maila desberdinetan aztertu dira eta esperimentazioa datu-multzo zabalago batekin egitea proposatzen da, anotazio automatikoa erabilita eta informatiboagoak diren ezaugarri linguistikoak erabilita.

Hitz gakoak: ezaugarri linguistikoak, itzulpen automatikoa, postedizioa, kalitatearen estimazioa, gomendio-sistema.

Abstract

The implementation of a machine translation system into production is not enough to warrant its efficient use. There exists the need to know when it is profitable to use machine translation as opposed to translating from scratch. That is why being able to estimate the quality of a machine translation is crucial. This thesis investigates the task of quality estimation of machine translation for a specific machine translation system and a specific domain by developing a recommender system for Spanish to English. The work further investigates how quality estimation can benefit from the use of linguistic characteristics in contrast to the more common shallower features. The data was collected from real translators who performed a post-editing task, and the linguistic features were manually annotated. First, we build a classification model that selects sentences for post-editing or translating. Secondly, we perform a regression task based on three quality indicators: Quality, Time and HTER. Although experimentation shows some promising results, overall the selected features are not discriminative enough for the recommender system to be implemented into production. Results are discussed at different levels, suggesting a replication at a larger scale, with automatic annotation of informative linguistic features.

Keywords: linguistic features, machine translation, post-editing, quality estimation, recommender system.

“The original is unfaithful to the translation.”

Jorge Luis Borges

The question this thesis aims to address is how unfaithful the translation is to the original.

Acknowledgements

The completion of this thesis would not have been possible with the support of my advisor Nora Aranberri. I would also like to give thanks to MondragonLingua and Vicomtech, who allowed me to use their resources. Finally, I also appreciate the constant support given to me by family, friends, and my partner.

Contents

List of figures

List of Tables

List of abbreviations

- CAT: Computer-Aided Translation
- EBMT: Example-Based Machine Translation
- LSP: Language Service Provider
- MT: Machine Translation
- NMT: Neural Machine Translation
- NTI: Negative Translatability Indicator
- PE: Post-editing
- QE: Quality Estimation
- RBMT: Rule-Based Machine Translation
- SMT: Statistical Machine Translation
- TM: Translation Memory

1 Introduction

The globalised world we are living in requires appropriate tools that suppress language barriers to enable communication among all of us. It demands translation tools. Moreover, due to the fast and ever-changing dynamics of our everyday life, it demands automated translation tools. Some examples of these are Google's Pixel Buds or Skype's Translator (both claim to provide real-time machine translation). However, when does the quality of an automated translation become good enough to be useful instead of cumbersome?

Machine translation (MT) is increasingly becoming more popular, with mainly two different goals (O'Brien, 2005): informational purposes (gisting rapid but imperfect messages) and publishing purposes (setting a starting point for professional translators). The latter goal demands high-quality machine translation, which is still a present-day unresolved issue (Fujita & Sumita, 2017). Therefore, Computer-Aided Translation (CAT), the use of computers for facilitating the translation process to professional translators (Bower & Fisher, 2010), such as Translation Memories (TM), is the current technology used by most companies in the translation industry (ISO/TC27, 2017).

Souza et al. (2015) state that a transition is taking place in the translation industry. Top corporations in the industry have already implemented state-of-the-art machine translation techniques in their companies's workflows, as the TAUS Machine Translation Market Report 2017 indicates (Joscelyn et al., 2017). The use of machine translation increases the work speed, which usually leads to an increase in productivity and hence, profitable gain. Nevertheless, we must take into account that MT is not flawless, since "the quality of automatic translations tends to vary significantly across text segments" (Shah, Cohn & Specia, 2015, p.101). Some of the questions that arise when considering this are: How to know when it is "worth" using MT? Can we measure how good a machine translation is? Furthermore, can we accurately predict how good a machine translation will be?

These are popular yet still unanswered topics which the task of Quality Estimation (QE) aims to solve. As Specia et al. (2010) state, the goal of QE is to indicate the quality of a machine translation, without access to reference texts, that is, texts produced by a professional translator.

In this thesis, we test an approach to the task of QE by designing a tool to estimate the quality of a machine translation based on the characteristics of its source text, to be able to decide when it is profitable to use MT. In all, the final aim of this research is to facilitate the decision-making process regarding MT in translation projects by providing an a priori indicator of the MT quality and classifying each segment for post-editing or translation.

The present work is divided into six main parts: motivation, theoretical framework, our approach, methodology, results, and conclusions. The motivation intends to clarify what the incentive and goal of this research are. The theoretical framework englobes the research context in which the study fits in. Our approach introduces a possible solution to the problem of QE. The methodology deals with the process undertaken to design the estimation tool and the experiments that were carried out. Next, the results present the findings of the experimentation. Finally, the conclusions contain the concluding remarks and ideas for future work.

2 Motivation

The decision to undertake this research was motivated by my involvement as an intern in a renowned Language Service Provider (LSP), MondragonLingua, a company who provides translation, localisation, and interpreting services. MondragonLingua is a translation company among the market leaders in Spain. My endeavour within the corporation was to investigate the way in which MT could be implemented and integrated in the existing workflow of the company, based on CAT tools and TMs, to improve the quality of the final product and be able to compete with the current technology used in the field.

The setup that propelled this study was, therefore, concrete: I worked with technical texts from the domain of elevators, with a neural machine translation (NMT) engine developed by Vicomtech and with the language pair Spanish-English.

After completing the implementation of MT, there was still a need to define which texts would benefit from MT and which would instead profit from translating from scratch. The overall attitude towards MT from the translator's view was disapproving, and therefore, it was important that the translator only received good quality MT not to further jeopardise their inclination to reject MT. In other words, there was an urge to be able to estimate the quality of the translation provided by that system to decide when to use MT as opposed to translate from scratch.

This thesis will address the issue of MT Quality Estimation (QE) by designing a classifier based on the source text’s linguistic features. The system will ideally recommend the user whether it is better to post-edit an MT sentence or if its quality is so bad that it is better to translate it manually.

3 Theoretical framework

This thesis aims to design a classifier based on the estimation of the quality of a machine translation provided by a specific MT system. To that end, we focus on the translatability of the source text based on its linguistic characteristics.

This section contains a summary of state-of-the-art machine translation systems, an overview of different current solutions to the problem of post-editing (PE) and quality estimation (QE), and recapitulates previous approaches to the task that provide the scientific proof that motivates our approach.

3.1 Machine translation

This overview of Machine Translation (MT) will provide the reader with the information needed to understand the technology used to produce our corpus of machine translated sentences.

MT can be defined as "the use of computer software to translate one natural human language into another" (U.S. Patent No. 9,798,720, 2017, p.14). It is opposed to CAT tools, which benefit from TM. A TM stores previously translated sentences for the translator to reuse when translating new content. Therefore, the human translator decides how to reuse the information provided by TMs, while MT represents an automatic translation technology (Somers, 1999).

Since the emergence of MT in the 1960's, there have been different approaches to the building of the translation model, leading to the different classification of MT systems that exist nowadays. We present each of them, together with their advantages and disadvantages, and their adequacy for the task of QE.

3.1.1 Theory versus data

The first distinction to be made concerning different kinds of MT systems regards the base of the engine (Sommers, 1999). On the one hand, there are theory-based techniques, which are based on transfer rules extracted from linguistic knowledge. These are called rule-based MT systems (RBMT). On the other hand, there are data-driven methods or corpus-based methods, which are based solely on massive amounts of data and do not require linguistic expertise. Example-based MT (EBMT), Statistical MT (SMT), and Neural MT (NMT) belong to this category.

Theory-based methods

The materialisation of MT came into being with Rule-Based Machine Translation (RBMT) in the 1960's. RBMT depends on expert's knowledge, as it is based on "explicit linguistic data such as morphological dictionaries, grammars, and structural transfer rules" (Forcada et al., 2011, p.128). The basic procedure to build such an engine is the definition of rules and features specific to a language to be able to analyse the input and generate an output.

Among the advantages of this kind of systems, as stated by Forcada et al. (2011), we find that RBMT allows having terminological consistency, which means that specific terms and words will always be translated in the same way (their translation does not depend on the context). Also, when performing an error analysis, errors are easier to diagnose and correct, as a specific rule probably causes them. Therefore, the MT output contains repeated errors easier to locate and hence, easier to fix by the professional translator. Furthermore, the linguistic data encoded in the rules for a particular language can be transferred to several target languages. Finally, RBMT also works exceptionally well for less-resourced languages, since it does not depend on available bilingual corpora, and for morphologically rich languages, since the rules allow to generate the correct morphological form of each word.

Nonetheless, RBMT is not the most used kind of MT systems. As it is based on the direct application of the transfer rules, the output translations it provides are entirely mechanical, less fluid and repetitive as opposed to newer systems (Forcada et al., 2011) (see 3.1.1.2 Data-driven methods).

There exist some Free Open Source (FOS) RBMT systems. Here I mention some. There is Apertium (Forcada et al., 2011), developed initially for Spanish-Catalan and Spanish-Galician translations, currently being extended to other language pairs (Catalan-English, Norwegian Bokmål–Nynorsk, Swedish–Danish, among others). There is also

OpenLogos (Scott & Barreiro, 2009), developed for English and German as source languages and French, Spanish, Italian and Portuguese as target languages. Finally, Matxin (Alegria et al., 2007) was initially developed for the Basque-Spanish language pair.

RBMT presents itself as advantageous for a QE tool since the direct application of rules makes it easy to identify structures that are hard-to-translate or vice versa. This would allow a more straightforward prediction of the quality of a sentence, as the technology behind RBMT is consistent and context-independent. However, as the research for this study is strictly related to the real setting in which it is conducted, this thesis does not cover RBMT output.

Data-driven methods

Due to the low performance of RBMT systems, data-driven methods, also called Corpus-Based methods, emerged during the 1980's. The emergence of this kind of MT systems was also motivated by the expanding power and storage capacity of computers, as well as by the growing availability of parallel corpora, driven by the digitalisation of text (Forcada et al., 2011).

Corpus-Based MT (CBMT) makes use of a database of aligned parallel texts, that is, a text paired with its translation into another language (Koehn, 2009). Since they do not depend on any linguistic knowledge, they are language-independent.

The general procedure followed by data-driven techniques for MT consists of matching new translation input against already translated sentences to get a suitable translation draft that may be reordered for the final proposal (Sommers, 1999). There exist several approaches that, although they share the basis, present different solutions to the problem. These are Example-Based Machine Translation (EMBT), Statistical Machine Translation (SMT) and Neural Machine Translation (NMT).

Example-based machine translation

The first Corpus-Based MT system appeared in the 1980's under the name of Example-Based MT (EBMT). This makes use of already translated sentences stored in a database.

To translate a sentence, EBMT finds parts of this sentence in the database, it extracts these bilingual phrases and, finally, it combines them to generate a new adequate translation.

Although it may seem similar to TM, EBMT is a proper automatic translation technique. While TM lets the human decide, EBMT decides automatically which translation match is the best (Sommers, 1999).

The advantage of this methodology is that it does not require any linguistic knowledge as it works solely from a database. However, an issue that poses a problem when dealing with this technique is the alignment of parallel corpora, that is, the matching of sentences that correspond to each other (Sommers, 1999). Also, an open question regarding EBMT is level at which it should work: at the sentence-, phrase-, or word-level. Moreover, the main disadvantage of EBMT is that it needs an extensive database, which means that it can only work with languages that have resources available.

Modern-day data-driven methods have discarded pure EBMT systems as new approaches to MT allow increasing its quality. Although Sommers (1999) developed an EBMT system and provided a thorough explanation, this kind of systems are no longer used in isolation. Therefore, this kind of MT systems was not considered for our study.

Statistical machine translation

Statistical Machine Translation (SMT) is currently widely used in the field of MT. It relies on both parallel and monolingual corpora to build a Translation Model and a Language Model, respectively. These allow the SMT system to assign a score to every possible translation of a given input sentence. The highest one is considered to be the best, and therefore, the final output.

The translation model contains a probabilistic score for each instance and their possible translations indicating how probable it is for that word to be translated as all the possible translations. Similarly, when applying the language model, every target sentence in the corpus is given a probabilistic score indicating how probable it is for that sentence to occur in text (Kohen, 2009). The bigger the language model, the more data the system will have and the better results it will provide.

SMT can work at different levels. There exist word-based models and phrase-based models (Kohen et al., 2007). Their functioning is the same and what changes is the unit they work on: words or phrases (any sequence of words).

As Forcada et al. (2011) state, SMT is state-of-the-art MT as it is known to provide fluent and natural translations. In contrast to RBMT, which is faithful to the original and provides consistency, SMT works more freely and may offer different translations for the same word, as it learns words in context. The main disadvantage is the one that all Corpus-Based models share: it needs a vast database of bilingual aligned sentences to be trained – although newer studies are exploring the possibility of building CBMT systems without parallel corpora (Artetxe, Labaka & Agirre, 2018; Søgaard, Ruder & Vulić, 2018).

Currently used FOS SMT systems are Moses (Koehn et al., 2007), which is a phrase-based model and cdec (Dyer et al.,2010) that allow the user to train SMT models at different levels. SMT provides fairly good translations and systems using this technique obtain state-of-the-art results.

In our study, we considered the use of an SMT system developed by Vicomtech, as it was available and it offered good MT results. However, after a thorough linguistic analysis performed on the output translation, we opted for an NMT engine because it offered results with higher quality in general.

Neural machine translation

Neural Machine Translation (NMT) is the newest approach to CBMT. It is based on the technology of neural networks and uses deep learning (Cho et al., 2014). NMT appears to be the future of MT, as it presents promising results.

Although different approaches are emerging for this, the first encoders were Klein et al. (2017), who provide a condensed explanation of how their NMT engine works. NMT functions as an encoder-decoder problem. Their encoder is a recurrent neural network (RNN) that vectorises each word of the source sentence. The decoder is also an RNN that takes into account previously translated words to predict the score of the next possible target word. Other methods propose the use of recursive convolutional networks instead of RNNs (Cho et al., 2014).

NMT presents a striking advance in the technology of MT. The use of neural networks allows NMT to use very little memory compared to SMT, which presents itself as a clear, practical benefit. Plus, the training of the translation model takes into account every component at once to maximise the results (Cho et al., 2014). Furthermore, it allows translations between two languages for which there is no parallel data available (Artetxe, Labaka, & Agirre, 2017). However, to obtain a neural translation model, the training is computationally expensive. Furthermore, the fact that NMT uses very little memory to function comes at the cost of a significant processing capacity, which limits the equipment with which NMT systems can be built.

The number of available FOS NMT system is recently increasing. Among them, we find OpenNMT (Klein et al., 2017), a complete toolkit to train and implement NMT models. Similarly, there is also Nematus (Senrich et al., 2017), developed by the University of Edinburgh. Finally, Artetxe et al. (2018) published UNdreaMT, the newest FOS toolkit for unsupervised NMT.

In this thesis, we use a MT system of this kind. More concretely, it is an attention-based encoder-decoder neural MT system (Bahdanau et al., 2014). Our MT engine offers good quality translations which presents itself as an advantage. Nonetheless, as we focus on the use of linguistic features for QE, the fact that NMT works as a black box makes it more difficult to identify straightforward causes for the errors found in MT output.

3.1.2 Hybridisation

Another current trend in the field of MT is the use of more than one MT system to obtain an optimal functioning. This is called Hybrid Machine Translation (Sawaf et al., 2017). There are several possible combinations to implement Hybrid MT, here we present the most recent ones.

Dhariya et al. (2017) present a combination of SMT, EBMT and RBMT to outperform the baseline of each of these methods individually for Hindi to English translations. The candidate translation chosen by SMT and EBMT is then passed to the RBMT system, which is available to generate more accurate structural and morphological constructions. In contrast, Dahlmann et al. (2017) present a hybrid system composed by phrase-based

SMT and NMT for German to English and English to Russian. They obtain good results as the MT quality improves by 2.3% BLEU in comparison to NMT alone.

As we have seen, some of the most recent MT systems present this architecture, which seems to combine the best characteristics of each technology to produce an optimal result. This kind of MT engine would also be worth exploring for a QE tool, as it would present high-quality translations, plus consistent results. However, again, this kind of system was unavailable for our study.

3.2 Post-editing and quality estimation

3.2.1 Post-editing

As stated in TAUS Machine Translation Market Report 2017 (Joscelyne et al., 2017), the implementation of MT in translation services is currently taking over the more traditional approach of using TMs and CAT tools. They claim that Post-editing (PE) is likely to replace the TM leveraging as the primary source of CAT in the next five years. PE refers to the task of manually correcting the errors of a translation produced by an MT system until a high-quality, publishable translation has been reached (Hokamp, 2017). The current implementation of MT takes the form of a “hybrid” text, obtained by the combination of matches from a TM, and machine translated sentences.

Research has been undertaken to investigate the benefits of PE as opposed to the classical translation supported by TMs. Arenas (2008) conducted an experiment with eight translators that post-edited and translated a text from English into Spanish. She proved that the task of post-editing MT texts increases productivity in contrast with translating texts from scratch up to a 25%. Plitt & Masselot (2010) also explored this field by setting an experiment with twelve participants on a translation from English to French, Italian, German and Spanish. Their results show that MT allowed translators to improve their productivity by 74%. A third study conducted by Federico, Cattelan & Trombetti (2012) in a similar setting reported productivity gains for all the twelve participant translators. In all, the advantage of the application of post-edited MT in the translation environment seems hopeful.

A still open research path concerning PE is how to measure PE effort, that is, the human effort involved in the task of PE. This effort may be divided into three (sub)spheres: temporal, technical and cognitive (Krings, 2001). The temporal effort is usually measured as the words/second rate at which a translator can translate, it is related to the PE time and traditionally seen as a combination of the technical and cognitive effort. Technical effort refers to the keystrokes and operations carried out during the post-edition. One way to measure it is to compute the minimum edit distance between the automatic translation and the post-edited version (Tatsumi, 2010). Finally, the cognitive effort involves the thinking process that a translator undergoes during the task of PE, which includes the identification of the errors of the machine translated sentence and the needed steps to correct it (Koponen et al., 2012). This kind of effort presents itself as the most difficult to quantify. There have been several studies that explore this issue, some of the proposals are based on Think-Aloud-Protocols, Choice Network Analysis, keystroke logging (O'Brien, 2005), pause ratios (Lacruz, Denkowski & Lavie, 2014) and human scores estimating the amount of PE necessary (Koponen et al., 2012).

Although the quantification of PE effort is a current question in the field of MT, this topic is beyond the scope of this thesis. Nonetheless, predicting PE effort is a task tightly related with Quality Estimation (QE). O'Brien (2011) found that there are reasonable correlations between automatic quality measures, such as TER (Translation Edit Rate), and actual PE productivity. Conversely, QE is also a popular research topic in MT.

Linguistic features for post-editing effort

Modern-day state-of-the-art research concerning PE is focused on the measurement of PE effort, still unresolved. There exists related work that makes use of linguistic features for estimating PE effort, which inherently is somehow related to QE. Two main projects investigated this approach.

Green, Heer & Manning (2013) show that Part of Speech (PoS) counts and syntactic complexity are predictors of translation time. More specifically, nouns have a significant effect on PE time estimation, and adjectives too, but less strongly. They work with English as the source language.

Research undertaken by Vieira (2014) identifies predictors of cognitive effort. In contrast to the results obtained by Green et al. (2013), source-text linguistic features show almost no effect on predicting PE effort, except a small significant effect of prepositional phrases. Vieira hypotheses that this is due to the difference of the source language since they work with French: “It is worth noting that nouns can act as direct modifiers of other nouns in English, whilst in French and other Western European Romance languages, this modification would normally require the use of prepositions.” (Vieira, 2014, p.205)

In all, the results of these two studies show that it is viable to predict PE time by providing linguistic features, which means that the characteristics of the source text affect the output of the automated translation. This idea will be explored further in our approach by estimating the quality of a text solely based on linguistic features.

3.2.2 Quality estimation

Quality Estimation (QE) refers to the task of Natural Language Processing that estimates the quality of a translation system without relying on reference, human, translations (Specia, Raj & Turchi, 2010). Although human assessment is known to provide the most thorough evaluation of MT, humans are also subject to inconsistencies (Graham, 2015). QE emerged from the necessity of an automatic method to assess the quality of MT systems. The rise of the importance of QE has motivated several proposals that approach this task; nevertheless, it is a present-day still open research topic due to the low accuracy of current MT engines (Martins et al., 2017).

The current approach to the task of QE involves using supervised machine-learning techniques to predict the quality of previously unseen machine translations (Shah, Cohn & Specia, 2015). These models are trained with a set of features, extracted both from the source text (sentence length, ambiguity, among others) and the automated translation (fluency, grammaticality), and when possible, from the MT system (scores, which somehow represent its confidence). These features are then trained along with a set of quality indicators, the training labels or classes. These may be anything that provides a pointer of the quality: PE time, an accuracy score, a fluency score… (Specia, 2013).

QE can be performed at different granularity levels: word-level, phrase-level, sentence-level and document-level estimation. The goal of word-level QE is to assign a label of OK or BAD to each word of the translation (Martins et al., 2017). The assignment of the labels is obtained by the alignment of the raw machine translation and the post-edited version of the training data. When words need to be edited, the BAD label is assigned. The same procedure is followed to predict QE at phrase-level. The goal of sentence-level quality estimation is to assign a label to a sentence, for example, based on the proportion of BAD words (Bojar et al., 2017). The goal of document-level quality is to assign a label to the translation of a whole document to indicate its quality. Until now, the best results have been achieved with sentence-quality estimation (Longcheva et al., 2016).

There exist some FOS toolkits that allow the implementation of QE task. Some of these are QuEST++ (Specia et al., 2015) for word-level and sentence-level QE (it will be used in this thesis to create a baseline set of features), and MARMOT (Logacheva et al., 2016), for word and phrase-level QE, however, it may be easily extended to the sentence level.

Applications

The use of QE for the task of PE has been proven to provide an increase in productivity (Specia et al., 2010). Translators can benefit from the inclusion of QE by gaining more information about the texts they are to post-edit. Instead of comparing the source text with the raw translation, the professional translators obtain a quantification of the translation quality upon which they can make decisions (Specia et al., 2010).

The most significant applications of QE from which LSPs can benefit are the following. QE may help discarding bad translations under a specific threshold, considering that the translation is not good enough to be post-edited and hence, it is easier to translate from scratch. Also, when several MT systems are available, the translation with the highest quality can be automatically selected and presented to the user. Finally, another gain is the possibility of estimating the human effort required to post-edit a text and therefore, to decide whether to use MT for that text or not. The tool developed in this research is intended for a particular application of QE: deciding whether or not to use MT at sentence-level.

Linguistic features for quality estimation

Previous research on the contribution of linguistic features to sentence-level QE has been undertaken by several studies that show consistent, albeit marginal, improvements in the results. Specia et al. (2011) included PoS tagging, chunking, dependencies and named entities for English-Arabic QE and obtained good results only in certain testing conditions. Hardmeier (2011) used constituency and dependency trees in a classification task for English-Swedish/Spanish QE. Their best results were achieved by combining traditional features with constituency trees.

The most recent publication on this matter was conducted by Felice & Specia (2012), developed for English-Spanish MT. Their proposal originates from the hypothesis that QE can benefit from the use of linguistic features extracted from the input and machine translated texts. This idea is based on the fact that evaluation of MT quality with reference translations has proven that when these metrics are enriched with linguistic information, they correlate much better with human judgement, mainly at sentence level. To build their QE model, they used a combination of system-independent features extracted from the source and the translated text. Of those, 77 are shallow features, and 70 are linguistic. Among the linguistic features, they take into account PoS, content and function words, named entities, disagreement, and unknown words, among others. Although their system does not outperform the baseline, their results conclude that linguistic information is in fact complementary to shallow features and should be strategically combined when building a QE system.

Our research is based on the findings by these three studies, specifically the ones provided by the last two, as they offer the scientific context for our study. Our research will approach the task of QE by using linguistic features, extracted solely from the source text for the language pair Spanish-English, which to our knowledge has not been explored yet.

Previous approaches to quality estimation

Translatability

In the past, the task of QE was approached as the task of predicting a text's translatability. The term machine translatability is defined by Gdaniec (1994) as "the suitability of a particular document for MT"(p.97). Similarly, Uchimoto et al. (2005) describe it as "a measure that indicates how well a given text can be translated by a particular MT system." (p.235). By how well a document translates, we mean that the fewer errors the MT system outputs, the better a document will translate. These errors are any necessary changes (deletions, substitutions, insertions or shifts) carried out by the professional translator during the task of PE.

Research by Tatsumi & Roturier (2010) proves that there exists a relationship between the machine translatability of a text and technical PE effort it requires. Furthermore, O'Brien (2004) discovered that in general, sentences that contain structures that present a potential problem for the MT system increase the PE effort. Conversely, we can also conclude that a document translates well or it is suitable for MT when it needs little PE effort as increased machine translatability tends to decrease PE effort (O'Brien, 2005).

Therefore, being able to estimate the translatability of a document seems relevant to indicate the quality of MT output, and, inherently, to avoid tedious PE effort, which would imply a decrease in productivity.

The general process to measure the translatability index of a text is to define a tool based on Negative Translatability Indicators (NTIs). "An NTI is a linguistic feature, either stylistic or grammatical, that is known to be problematic for MT" (O'Brien, 2005, p.138). Three works investigated this approach: Gdaniec (1994), Bernth & Gdaniec (2001), and Underwood & Jongejan (2001). The steps to design such a tool are the following, as described by Underwood & Jongejan (2001): (1) identification of NTIs, (2) assignment of penalties or weights to each indicator, (3) computation of the translatability index.

The most common NTIs considered in the works mentioned may be classified into the following grammatical categorisation:

- Syntax: coordination, dependent and relative clauses, complement sentences, missing subjects, the passive construction, prepositional phrases, nonfinite verbs, verbless sentences
- Lexicon: out of vocabulary words, certain ambiguous words (-ing, as, with, …), homographs
- Morphology: part-of-speech ambiguity
- Semantics: time references
- Punctuation: parentheses, lack of initial capitalisation of a segment, missing hyphen
- Segment length: too long or too short sentences

Controlled Languages

Another research topic in the field of MT is how to improve the translatability of a given text in order to get a better translation output. The main idea to improve the translatability of a text that will be machine translated is to modify the source text itself.

Controlled Languages (CL) take this approach. “A CL is a form of language with special restrictions on grammar, style, and vocabulary usage” (Bernth & Gdaniec, 2001, p.196). These constraints aim to improve consistency, readability, translatability, and retrievability. Hence, if we apply CL constraints to a text, we will improve its machine translatability.

These approaches interest our study because of two main reasons: on the one hand, the NTIs found in the three publications above are very similar to the linguistic features we use in this thesis for QE. On the other hand, the assumption lying behind the concept of CL is that the characteristics of the source text have an impact on the MT output, as CL changes the source text and expect an improvement in the MT output.

4 Our approach

As we have seen, there are several approaches to the problem of QE for machine translation. All solutions to the issue of QE share the common ground of not using any reference texts. Furthermore, feature selection for optimal QE is considered to be one of the most challenging aspects of the task (Shah et al., 2015). Most works extract features from the characteristics of the source text and the MT output. In this aspect, our research explores a novel approach. On the one hand, it introduces a new perspective by not using the raw machine translations, which may allow QE from one source language to different target languages, and independently of the development of the MT system. On the other hand, it fills a gap in the field of QE by testing a different strategy regarding the features themselves.

The source text plays a significant role in our research as it is on what our QE tool depends. Linguistic features will be extracted from the source texts, similarly to Felice & Specia (2012) and the NTIs identification provided by Gdaniec (1994), Bernth & Gdaniec (2001), and Underwood & Jongejan (2001).

The visited works regarding translatability have taken into account potential problems that may occur in each of the three steps a translation transfer system consists of, namely, (1) source analysis, (2) transfer, (3) target generation (Bernth, 1999), or they are based on purely statistical properties that require no deeper syntactical or semantical analysis of the text. Nonetheless, after this overview of the current state-of-the-art proposals, we have decided to approach the topic by designing a tool solely based on the characteristics of the source text. The motivations of this decision are several. First, as LSPs tend to work from one source language to many target languages, we wanted to investigate whether features from the common source are valid for the QE task in an attempt to find shared indicators for all. Second, we also believe that focusing on source text features disassociates the link between the MT development and the QE models to a certain degree. Also, we want to emphasise the linguistic features of the original text to investigate what impact these may have on the quality of a document, as there exist previous evidence supporting this choice.

Tatsumi & Routier (2010) investigated the existing relationship between source text characteristics and PE effort. For their future work, they proposed the inclusion of these characteristics (linguistic features from the source text) in a translation recommendation system. Such a system would provide an indication of the translatability of a text based on its source characteristics, which is what we intend to do in this thesis.

Plus, we know that the earlier in the process the problems arise, the more likely they are to be carried and affect further steps (Bernth, 1999).This means that the NTIs found in the first step of the transfer (in the source analysis) are the ones that carry the most weight in the final translation. Nevertheless, we must take into account that this assumption was made for an RBMT engine and has not been studied with other kinds of MT systems.

The proposal of using CL to improve the translatability of a text implies the modification of the source text itself. Whenever we make a change in the text, this will influence the output of the translation. Conversely, we may claim that the source text carries an essential weight as it can affect the translation.

Finally, as we have previously seen, the inclusion of linguistic features for PE effort prediction (Green et al., 2013; Vieira, 2014) as well as for QE (Felice & Specia, 2012) shows that it is viable to investigate the impact these may have on the quality of MT. This also sustains our decision, since it means that the linguistic features of the source text somehow “hide” the key to quality.

5 Methodology

The primary goal of our research is to design a sentence-level recommender system based on QE for MT using linguistic features. To achieve that, we define linguistic features to annotate a corpus and also obtain different quality indicators. Then we train various machine learning models to predict the quality of the MT output.

In this section, we first describe the database used for this study. Then, we present the quality labels used for QE along with their annotation process. Finally, we discuss the identification and annotation process of the linguistic features and the experimental setting.

Annotation represents a big part of this project since the linguistic features, and the quality ratings were annotated manually. Three professional translators performed the annotation task providing the necessary data for our quality labels, and the author of this thesis performed the annotation of the linguistic features.

5.1 Database

The database used in this thesis is influenced by the fact that the experimentation took place in a real scenario. Industry collaborators provided the data and software, and hence they are subject to specific terms and conditions. As stated by Bernth & Gdaniec (2001), the quality of an MT output depends on three main factors: the MT system, the language pair and the domain. These factors then were given by the context where our project took place and, hence, the scope of our research is limited.

Regarding the MT system, we used the engine that reported fewer errors for the intended corpus, which is the one currently implemented by the LSP as part of their production. This is an attention-based encoder-decoder neural MT system (Bahdanau et al., 2014) developed by Vicomtech and customized for our client.

Concerning the language pair, in contrast with the previous research studies that focused on English as the source language (Felice & Specia, 2012; Handmeier, 2011), my work focuses on Spanish as the source language and English as the target language. The LSP's work requirements determined this choice. Spanish is a more complex language in terms of morphology and grammar than English. Therefore, we hypothesise that, the more hard-to-translate structures found in a Spanish text, the more influence they will have in the quality of the MT output translation.

The domain that was considered for this thesis belongs to the field of technical texts related to the area of elevators since the LSP owns a significant corpus of this domain provided by one of their major clients. These texts consist of installation and maintenance documentation of elevators. They include lots of repetitions, lists, enumerations, among others. Because of the annotation effort required, only a fraction of this corpus was used. The subcorpus annotated for this study consists of 7 texts and 6,542 words (See Table 1).

Text ID	**# words**	**# sentences**	**avg. word/ sentence**
1	360	25	14,40
2	663	76	8,72
3	444	53	8,38
4	726	70	10,37
5	1198	95	12,61
6	246	17	14,47
7	2905	174	16,70
Total	6542	511	12,83

Table 1: Description of the corpus

5.2 Quality indicators

To predict the quality of the afore-mentioned 511 sentences, three distinct quality labels are collected. Nonetheless, to decide what quality indicator our model predicts, first we need to define how quality is understood in this thesis. As the sentences that will be machine translated will later be post-edited by professional translators, our definition of quality for this research is related to PE. Therefore, a high-quality MT sentence is one that needs a minimum number of PE changes to transform it into a publishable translation.

Related work at sentence-level QE has provided different metrics to label each instance. For PE purposes, human scores have been considered (Specia et al., 2010; Felice

& Specia, 2012; Shah et al., 2015). These allow the user to choose among different options within a numeric scale to annotate each sentence. Nonetheless, the official label for sentence-level QE reported by the different participant teams at the shared tasks of WMT (Bojar et al., 2017) is the HTER (Human-Mediated Translation Edit Rate). We will use both measures as quality indicators.

As explained by Snover et al. (2009), the HTER measure is obtained indirectly by human annotators. They generate a new reference translation for the MT output more faithful to the original in terms of fluency and meaning. Then, this newly generated sentence is used as a reference for computing the Translation Edit Rate (TER). TER measures the number of edits (insertions, deletions, substitutions and reorderings) that a human performs on MT output to match a reference translation (Snover et al., 2006). Although HTER is time-consuming and regards all errors in the same way (it makes no difference between severe errors and minor edits), it is suited for our purposes. It specifically measures the minimum number of changes needed to achieve a good translation from an MT system.

Human scores provide a direct indication of the quality of the translation based on the amount of PE needed. We follow the methodology used by Lacruz, Denkowski & Lavie (2014), who ask the participant to rate a segment's suitability for PE after having performed the post-edition, according to the following 1 to 5 scale in Table 2.

Rating	**Criterion**
1	Gibberish – The translation is totally incomprehensible
2	Non-usable – The translation has so many errors that it would clearly be faster to translate from scratch
3	Neural – The translation has enough errors that it is unclear if it would be faster to edit or translate from scratch
4	Usable – The translation has some errors but is still useful for editing
5	Very good – The translation is correct or almost correct

Table 2: Rating scale for quality of MT

Finally, the PE time is recorded automatically and used as a third quality indicator. This further indicates the severity of the errors as we assume that, the more time invested in a sentence, the more difficult it is to correct its errors. This assumption is motivated by a study undertaken by Koponen et al. (2012) which confirms that PE time may be a good measure to estimate cognitive effort. Their results show that shorter PE times are linked to errors that are easier to correct.

Despite this, the aforementioned indication of quality is relative (Gdaniec, 1994). Therefore, for our ultimate goal that intends to predict whether to use MT or not, further treatment of the data is performed. This consists of the definition of a threshold, that will indicate if the sentence is to be machine translated or not.

To sum up, the data collected for each segment includes the Spanish source segment, machine translation into English, its English post-edited version, its annotation based on linguistic features, an HTER score, a human Quality score indicating the usefulness of the MT for PE and the PE Time.

5.2.1 Quality indicators for post-editing effort

Following the line of Koponen et al. (2012), the three aspects of PE effort defined by Krings (2001) are assessed: technical, temporal and cognitive (See Figure 1). Moreau & Vogel (2014) explore the limits of QE and also experiment with these three measures as quality measures. The HTER is used as a measure of technical effort, the PE Time as a measure of temporal effort and, finally, human Quality scores as a measure of cognitive effort.

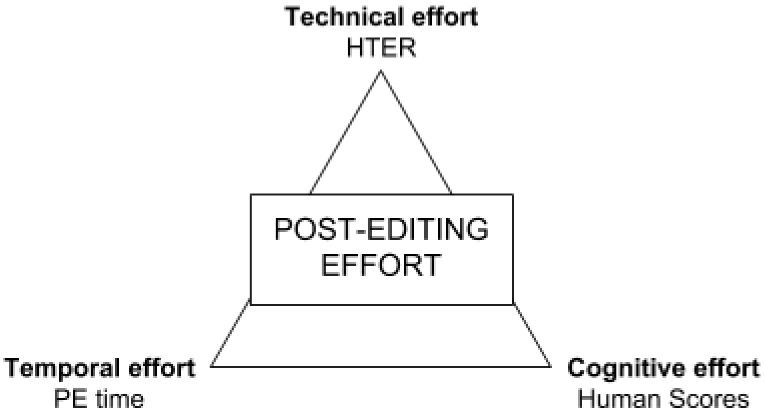

Figure 1: Measures for the three post-editing effort dimensions

5.2.2 Post-editing and Quality annotation tasks

We collected our MT quality indicators in a PE task performed by three professional translators. Each of them post-edited a subset of our corpus and rated the quality of each sentence. The two tasks were performed at once using an online platform for CAT named Matecat (Federico et al., 2014). To carry out the tasks, they were provided with the source and the MT sentence. They had to post-edit the MT sentence to achieve a good quality translation and, furthermore, rate the quality of the MT sentence by providing a score following the 1-to-5 scale showed above.

MondragonLingua put at our disposal three professional in-house translators for a limited amount of time, which led to the limited size of our corpus. The annotation data was split into three parts based on text boundaries and number of words and each translator post-edited one. In the end, all annotators had roughly the same amount of words (2180 on average) to post-edit (see Table 3).

Translators were given specific guidelines to ensure consistency during the PE process (see Appendix B). Furthermore, they were asked to fill in a survey about their background and experience in PE (see Appendix C).

All our translators are females, born between 1987 and 1991 and have been working in the translation industry from 2 to 4 years. They are all scientific, technical, and literary translators working with English and Spanish as their language pair. Regarding PE, they all had experience in post-editing machine translated sentences. However, their attitude towards the use of MT for PE is somewhat negative and claim that translation from scratch is better, as it is easier and faster. Concerning the specific PE task they undertook, they considered that the MT was not useful, although one claimed it was pretty accurate. According to this, translators' opinion on the use of MT for PE is clearly inclined towards a negative trend. Nonetheless, the average quality they gave to the MT sentences is of 4,6 within the 1 to 5 scale, which is quite high. We can see a summary of their performance in Table 3.

Translator	# texts	# sentences	# words	time	sec/word	sec/sent	HTER	quality
Trans 1	4	224	2168	02h:07m:16s	3,5	34	6,83	4,66
Trans 2	2	150	2173	02h:34m:30s	4,3	62	14,90	4,56
Trans 3	1	137	2198	04h:59m:53s	8,2	131	10,14	4,66

Table 3: Translators' performance in the post-editing task

We will examine the work of the translators individually. Translator 1 had the most varied set of sentences, as it included sentences from four different texts. The average time spent per sentence is of 31 seconds (0,57 min). This translator is the fastest. She is also the one that modifies less the provided translation, as her average HTER is the lowest, it is 6,83. The average quality of her quality ratings is 4,65.

Translator 2 had sentences from two different texts. The average time spent per sentence is of 1 min. Her average HTER is of 14,90. She is the translator that changes more the sentences during the post-edition. The average quality of her quality ratings is 4,56. She is a bit stricter with the quality ratings of the sentence than the other two translators.

Translator 3 had sentences from only one text. The average time spent per sentence is of 2,2 min. She is the slowest translator. Her average HTER is of 10,14. Coincidentally, the average quality of her quality ratings is exactly the same as Translator 1: 4,65.

We can observe how, regardless of the time spent or the number of texts, the quality scores are consistent among all three translators.

It is important to mention that 1% of the sentences (5 sentences in total) had missing values for the quality indicators. The missing values were replaced by the median of the total values (5).

Dataset description

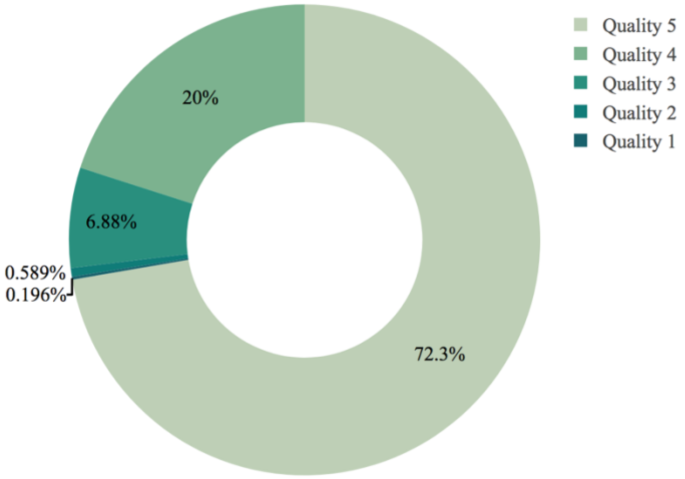

Figure 2: Distribution of sentences according to quality

In Figure 2, we see the distribution of the 511 sentences according to the label provided by the three translators. A significant part of sentences (up to 72%) are labelled as being of quality 5, which means *Very good – The translation is correct or almost correct.* This indicates that, overall, the MT system chosen was quite good. The 20% of the sentences were labelled as 4 and 7% as 3. Labels 2 and 1 were practically not used. The fact that each sentence was annotated only by one translator may bias these results.

Quality indicators correlations

To inspect the relationship between the three quality indicators – Quality, HTER, and Time, we have performed an inspection of the results at different levels.

We have calculated the Spearman correlation among the three classes. The results are shown in Table 4.

Quality indicators	(1)	(2)	(3)
(1) Quality	1.0000		
(2) Time	- 0,3856*	1.0000	
(3) HTER	- 0,7171*	+ 0.4435*	1.0000

Table 4: Spearman correlation of quality indicators (p < 0.001)*

As we can observe, Quality and HTER have the strongest correlation, achieving a strong correlation. However, the results of correlation with Time are lower both for Quality and HTER. Quality and Time have a weak correlation, while HTER and Time have a moderate correlation, the only positive one.

To further inspect the relationship among the three quality indicators, we can observe the following three graphs.

In Figure 3 we can observe the negative correlation between Time and Quality. The higher the Quality score, the lower the Time. This fits into our understanding of PE effort, as we assume that the best MT require less time to post-edit. The box for Quality score 5 is comparatively short; this suggests that overall sentences labelled by the translators as having a Quality of 5 were post-edited in a more similar time range. On the contrary, the box for Quality score 3 is comparatively tall, which indicates that the PE time varies the most.

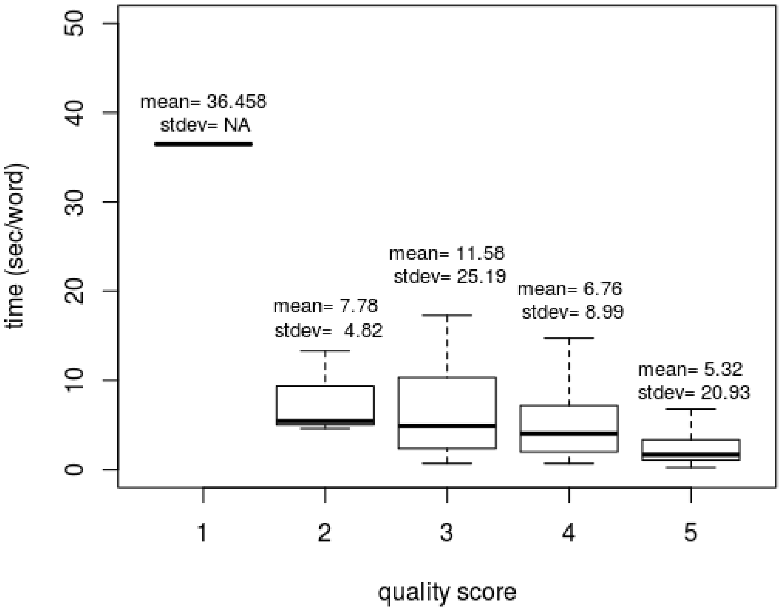

Figure 3: Correlation between Time and Quality

It is worth mentioning how the time decreases for Quality score 2. It may be because when a sentence is labelled as 2, this implies that translating from scratch is faster than PE. Probably the translators deleted the whole sentence and created an entirely different one.

In Figure 4 we can observe more clearly the negative correlation between HTER and Quality. For quality score 5, the box is practically non-existing, and that is because most sentences labelled as 5 have an HTER of 0, as 5 indicates that the sentence is flawless.

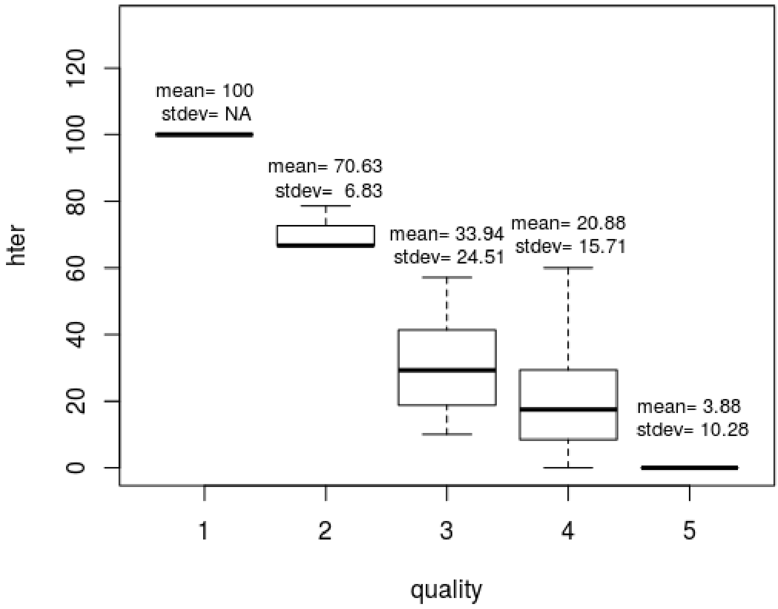

Figure 4: Correlation between HTER and quality

We can now confirm what we just mentioned about Quality score 2. For Quality score 2, the HTER is the second highest. This means that sentences where heavily modified and backs up our hypothesis that when sentences are labelled as 2, the translators may have deleted the machine translation and started from anew.

Lastly, in Figure 5 we can see how the correlation between HTER and Time is positive. Although most instances are found with lower HTER and lower TIME, there is a tendency that indicates that the higher the HTER, the higher the time the translators take to post-edit a sentence. This supports our understanding of PE effort, as we assume that the more edits (insertions, deletions, substitutions and shifts – represented by HTER), the more cognitive effort it takes (represented by the time).

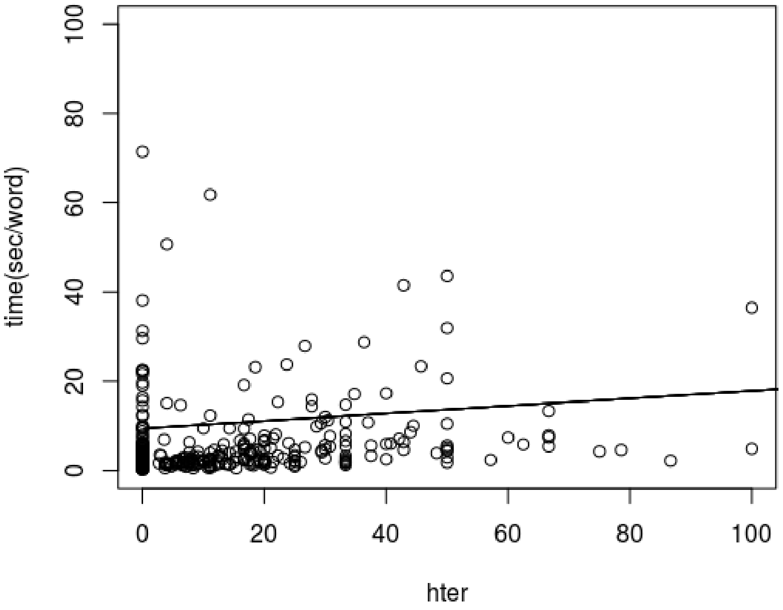

Figure 5: Correlation between HTER and time

Quality indicators for machine learning

The ultimate goal of this thesis is to build a recommender system that will discern which sentences should be post-edited and which should be translated from scratch. That is what our classifier will aim to do.

For the binary classification, we need to establish a threshold to convert each of our quality indicators into the Post-edit (PE) or Translate (T) class. This is usually done by comparing the average translation time per word as opposed to the PE time per word obtained using translation and PE tasks (Aranberri & Pascual, 2017). Other approaches are based on a specific HTER score provided by WMT for establishing a threshold. However, as we do not have these measures, the threshold is calculated based on the results of the PE task performed by the three translators.

Quality > 3

Table 2 shows the 1-to-5 quality scale according to which the translators had to rate each MT sentence. High quality scores mean high quality. Scores 4 and 5 followed this description:

- 4: Usable – The translation has some errors but is still useful for editing
- 5: Very good – The translation is correct or almost correct

Scores 3 and below meant it was not clear or it was better to translate from scratch. Parting from this definition, we establish the threshold in 3. Quality scores 4 and 5 are post-processed into PE and scores 1,2,3 are post-processed into T.

Time < 11 sec/word

Time is recorded in miliseconds (ms) and the lower the time is, the better quality a sentence has. The average PE time for the sentences of our dataset is of 6071 ms per word, that is 6,8 seconds per word. As we have said, for quality scores 4 and 5, it is clear that PE is better than translating from scratch. If we have a second look at figure 4, we can see how the average time for quality score 3 is of 11,58 sec/word. We assume that sentences that have a score of sec/word lower than 11,58 sec/word belong to 4,5 quality scores, for which it is better to post-edit than to translate. Therefore, we set the threshold at 11 sec/word.

Furthermore, MondragonLingua works under the assumption that the average time for translation tasks in the company is of 2,500 words per a 8-hour workday. This translates into 313 words per hour and 11 sec/word. This coincides with the threshold we established of 11 sec/word and reinforces the idea that if a sentence has a score higher than 11 sec/word, this should be translated from scratch. If time is smaller than 11 sec/word, the sentence is post-edited. If time is higher than 11 sec/word, the sentence is translated from scratch, as we assume that it is faster to translate from scratch than to post-edit it.

HTER < 33

HTER represents the number of edits performed during the PE process. Therefore, the lower the HTER score, the better the quality of the MT output. To set a threshold for HTER, we look at Figure 5 and see that for quality score 3, the average HTER is 33. Thus, we set the threshold at 33 and assume that sentences with an HTER lower than 33 are of sufficiently good quality as to be selected for PE.

We have established thresholds for each quality indicator. However, we need to bear in mind that these thresholds are established after a post-edition and quality rating task. So they are deduced from an initial PE work that needs to be performed in order to decide where to set the thresholds.

Distribution of sentences for each quality indicator

We applied the thresholds described above to our dataset to transform our quality indicators into binary classes: PE and T. For Quality, we join categories 1,2,3 into T and 4,5 into PE. For time, those sentences that have a time above 11 sec/word are converted into T, the rest for PE. Finally, those sentences with an HTER above 33 are assigned the label T and the other ones, PE.

Class	**Quality**	**Time**	**HTER**
PE	470 (92.34%)	463 (90.96%)	456 (92.34%)
T	39 (7.66%)	46 (9.04%)	53 (10.41%)

Table 5: Distribution of sentences into PE and T per quality indicator

In Table 5 we can see the distribution of these newly defined classes regarding sentences and their corresponding percentages. We observe that the class T only represents 9.04 % on average out of the total number of sentences. One might be tempted to think that given the low percentage of source sentences to be allocated to the translation class identifying this might not be worth. However, in a production setup, where optimum productivity is sought and translators’ perception plays such a key role, filtering out these segments might prove essential.

For our classification task, each quality indicator will be trained on their own data set provided by the three different thresholds. We carried out experimentation in this way for simplicity, but further research into finding the optimum threshold is worth doing.

5.3 Linguistic features

To establish the features that affect the translatability of a document negatively, the texts that make the final corpus used in this study have been manually analysed. Since the source language (Spanish) differs from the one used in previous studies (English), the linguistic features were identified from scratch. The process followed consisted of a comparison of the source text, the generated automated raw translation and the reference text (written by an in-house professional translator). Whenever the author found a disagreement between the automated translation and the reference text, she would go back to the source text and try to identify a structure or a pattern that was the cause of the discrepancy by means of a linguistic analysis. As the chosen MT system was built on a neural architecture, this process was not always straightforward. An example of the process followed to identify the linguistic features is shown in Table 6.

- **Source**: [*Peligro de* [*levantamiento* [*manual de* [*objetos pesados*]$_{NP}$]$_{NP}$]$_{NP}$]$_{NP}$.
- **Reference**: *Danger of manual lifting of heavy objects.*
- **MT**: *Manual heavy object lifting hazard.*

Table 6: Example of identification of linguistic features

In the source sentence, there are four noun phrases, these may be the cause for the word order error in the MT output. There is also the polysemic word *peligro* that has been translated to *hazard* instead of *danger* → short noun phrases and polysemic words are selected as candidates to be linguistic features.

After a first look at the texts and their translation, 30 preliminary patterns were identified. Secondly, a more in-depth analysis of the features was carried out. It concerned (1) the severity of the generated problem - for example, the author would consider a syntactic error more severe than a punctuation error since it would hinder considerably more the understanding of the text. She also took into account (2) the ease of the feature to be identified - patterns that consist of specific words, for example, pronouns or determiners, require no linguistic resources, while identifying a noun phrase would require a morphological analyser.

Moreover, the features used in previous works defining NTIs (Gdaniec, 1994; Bernth & Gdaniec, 2000, 2001; and Underwood & Jongejan, 2001) were also considered to enter the list. The last revision left 28 patterns, which are the current linguistic features we will used to develop our tool, by annotating the corpus. Note that the patterns were used to annotate the whole corpus, not only when they caused errors.

We have defined two sets of linguistic features: a scalar feature set (24 features) and a binary feature set (3). The first one counts how many times a feature appears in a sentence and the second one produces a 1 if the feature is present and a 0 if it is not. Furthermore, sentence length has also been included as a feature, which gives us a total of 28 features for our experimentation. Next, a list of the final features together with an example is shown.

5.3.1 Linguistic features list

Scalar features

- long_np: noun phrase with more than one complement.
    - *[El **tamaño** [máximo] [de abertura de la malla]]*
- short_np: noun phrase with zero or one complement.
    - *[El **código** [del aparato]]*
- top_np: noun phrase that is at the top syntactic level and does not depend on any other noun phrase.
    - *[la **hilera** de [guías de [la losa de [la sala de [máquinas.]]]]]*
    - In this sentence there are 5 noun phrases, but only one top noun phrase.
- adjp: adjective phrase.
    - *el polipasto [**eléctrico**]*
- pp: prepositional phrase.
    - *Limitador [**en** el lateral izquierdo.]*

- top_pp: prepositional phrase that is at the top syntactic level and does not depend on any other prepositional phrase.
    - *el borde [**del** agujero [de la losa [de sala [de máquinas.]]]]*
    - In this sentence there are 4 prepositional phrases, but only one top prepositional phrase.
- advp: adverbial phrase.
    - *aplicarle una fuerza de 300N distribuida [**uniformemente**]*
- long_vp: verb phrase with more than one complement.
    - *mantener [la puerta] [en posición de cierre] [mediante dispositivos eléctricos de seguridad.]*
- nonfinite_v: verb in nonfinite form (infinitive, gerund, participle).
    - ***actuando** junto los dispositivos de seccionamiento automático contra choques indirectos.*
- finite_v: verb in finite form.
    - *El foso **será** accesible*
- dep_cl: dependent clause.
    - *El hueco no debe utilizarse para ventilación de recintos [que no pertenezcan a la instalación de ascensores.]*
- ellipticsubj_vs: subject that does not appear explicitly (Spanish is a pro-drop language) or inversion of the order SVO (subject-verb-object)
    - [Se indican] [requerimientos y recomendaciones para cada tipo de construcción de hueco.]
- se_particle: Spanish pronoun *se*.

  - ***se** instalarán paneles o placas de material imperforado entre la malla y el hueco*

- personal_pr: Spanish personal pronouns.
  - *En la placa Master hay un pulsador llamado de Conservación para hacer el test de voz y debajo de **él**, un led L8.*

- def_art: definite articles (*el, la, los, las*).
  - *Montar **las** patas de apoyo a **la** armadura*

- coor: coordination performed by copulative or disjunctive conjunctions.
  - *Cada conector tiene su manguera **y** función distintas **y** no intercambiables:*

- num_seq: sequence of numbers or numbers and letters.
  - *Puede reproducir más de **160** mensajes en cada uno de los **2** idiomas que tiene grabados.*

- abbrev: abbreviations of words
  - ***EXT**4.0 (**BYP**)*

- oovw: out of vocabulary words, includes abbreviations.
  - *POLIPASTO EN LA SALA DE MÁQUINAS, **TIRAK** EN EL PISO INFERIOR*

- neg: negation
  - *Las superficies de paredes, suelos y techos de los huecos deben ser de materiales duraderos y **no** propensos a la producción de polvo.*

- poly_words: polysemic words, words with more than one sense in Wordnet.
  - ***Describir** el **proceso** de **montaje** de las medidas compensatorias en los ascensores // con **altura** última **planta** reducida.*

- dom_words: words specific to the domain, contained in a glossary.

- *Instalación de **cable múltiple** adicional*

- sym: symbols.
    - *© Copyright*
- punct: any punctuation sign, except the ones contained within number sequences.
    - *Los GSM**,** OMU**,** etc no se configuran distinto**.***

Binary features

- no_verb: the sentence has no verb.
    - *Código instalación.*
- no_stop: the sentence has no full stop at the end.
    - *MONTAR LOS DOS PRIMEROS PUNTOS DE FIJACIÓN DE GUÍAS*
- long_sentence: the sentence is longer than 25 words
    - *Para ello, seguir la instrucción 0905011 "INSTRUCCIONES PARA KIT MÁQUINA Y VIGA DE TIRO ", apartado 2.4.3 "CAMBIO DE LOS PARÁMETROS DEL VARIADOR DE FRECUENCIA".*

5.3.2 Linguistic features in related work

As mentioned above, Felice & Specia (2012) conducted a similar study to estimate the quality of MT parting from linguistic features. Although they worked with the language direction en-es instead of es-en and the features of this thesis were identified from scratch, it is remarkable how much our features coincide with most of their proposed linguistic features. Here I will examine them more deeply.

They take into account content words (N,V,ADJ) in contrast to function words (DET, PRON, PREP, ADV). These PoS are also considered in our patterns. Also, Felice & Specia (2012) focus on explicit and implicit pronouns, in this research called "elliptic" pronoun. Finally, they care about unknown words (out of vocabulary words) using a spell checker.

Nonetheless, there are some features that this research does not contemplate. On the one hand, there are deictic elements, named entities and split contractions in Spanish (i.e. al = a el), not considered here because they were not targeted as hard-to-translate structures by the MT system. On the other hand, subject-verb disagreements that do not occur in the source text as they are grammatically correct documents. Regarding their proposal on shallow features, they take into account sentence length, unique tokens and numbers, non-alphabetical tokens, average token frequency, among other metrics. Among these, this thesis only considers sentence length.

In general, our features match most of the characteristics proposed by Felice & Specia, besides adding many more. The ones proposed by them that are not used in this study could be included for further research.

5.3.3 Annotation of linguistic features

The annotation of linguistic features was performed examining each sentence and annotating whether it contained or not each of the aforementioned linguistic features. Appendix A shows an example of an annotated sentence.

It is interesting to gather the results of the annotation to learn more about the corpus itself. As we can see in Table 3, most features appear consistently in the corpus, especially the presence of noun phrases, prepositions, “se” particle. However, other features appear less often, this is the case of negation and symbols that barely have any representation in the corpus.

Linguistic feature	Frequency	Percentage
short_np	1941	15.50%
poly_words	1479	11.81%
pp	1367	10.91%
top_np	1089	8.69%
def_art	912	7.28%
top pp	826	6.59%
punct	738	5.89%
dom_words	685	5.47%
adjp	491	3.92%
nonfinite_v	362	2.89%
finite_v	321	2.56%
long_vp	236	1.88%
dep_cl	234	1.87%
num_seq	213	1.70%
coor	213	1.70%
long_np	196	1.56%
oovw	191	1.52%
no_verb	188	1.50%
abbrev	157	1.25%
no_stop	153	1.22%
ellipticsubj_vs	142	1.13%
advp	123	0.98%
se_particle	106	0.85%
long_sentence	63	0.50%
personal_pr	48	0.38%
symbols	31	0.25%
neg	20	0.16%

Table 7: Distribution of annotated linguistic features

If we look at the results of the linguistic annotation in Table 7, we can see how the sentences being studied contain lots of nouns and prepositions, as well as many embedded long noun and prepositional phrases (if we observe the difference between long_np+short_np and top_np, just like the difference between pp and top_pp). There are also quite a lot of adjectives and much fewer adverbs. There is quite a high number of definite articles and a small amount of number sequences.

Regarding verb phrases, there are more or less the same amount of finite and nonfinite verbs and a 34% of the verb phrases have more than one complement (236 long_vp/683 total vp – 362 nonfinite_vp + 321 finite_vp). Dependent clauses represent a similar percentage. There are fewer elliptic subjects, “se” particles and coordinations. We can derive that the sentences in our corpus consist of mostly SVO structures and in general do not contain many complements nor dependent clauses.

Concerning the vocabulary of the corpus, we can see how there is a small presence of abbreviations and out of vocabulary words. More interestingly, many words belong to the domain and an abundance of polysemic words. If we take into account the total words of the corpus (6,542), polysemic words represent 22,6%.

5.4 Experimental setting

5.4.1 Dataset

The dataset used for all experiments consists of 509 Spanish sentences. Initially, there were 511 sentences, but two instances were identified as outliers and removed from further analyses.

5.4.2 Feature sets

We describe three groups of QE feature sets: the first group (BASE) consists of 17 "black box features" that have been extracted using the open source toolkit Quest ++ (Specia et al., 2015). They are 17 shallow MT system-independent features:

1. number of tokens in the source sentence
2. number of tokens in the target sentence
3. average source token length
4. LM probability of source sentence
5. LM probability of target sentence
6. number of occurrences of the target word within the target hypothesis (averaged for all words in the hypothesis - type/token ratio)
7. average number of translations per source word in the sentence (as given by IBM 1 table thresholded such that prob(t|s) > 0.2)
8. average number of translations per source word in the sentence (as given by IBM 1 table thresholded such that prob(t|s) > 0.01) 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 (SMT training corpus)
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 (SMT training corpus)
16. number of punctuation marks in the source sentence
17. number of punctuation marks in the target sentence

This feature set will work as our baseline, as it usually is for WMT QE shared tasks.

The second group (LING) contains the linguistic feature set described in this thesis that consists of 28 manually annotated linguistic features. Finally, the third group (BALI) consists of the two previous feature sets merged, 28 linguistic features + 17 baseline features.

5.4.3 Experiments

To study the task of QE in depth, we performed two distinct supervised machine learning tasks on the three feature sets. We used Weka version 3.8.2 (Hall et al., 2009) to perform the classification task (and later a number of regression tasks, see Section 6.2) with five different algorithms. Each algorithm was tested on the three different quality indicators (Quality, Time and HTER).

There exist many different regression and classification algorithms. For this task, we have selected five of the most used ones to explore the three feature sets.

- Logistic regression (L): maps values between 0 and 1 by means of the logistic function. It is used solely for classification. For the regression task, a Linear Regression (LR) algorithm will be used. It predicts values for the line that fits best the training data.
- Multi-Layer Perceptron (MLP): uses deep neural networks for both regression and classification.
- SMO: implements the Support Vector Machine for classification. SMOreg is the implemention for regression. In both cases, we used an RBF kernel, as it was proven to provide the best results in the study performed by Moreau & Vogel (2014).
- IB1: it is a nearest-neighbour classifier. It finds the k most similar training patterns to the test set to perform a new prediction.
- Bagging: also called Bootstrap Aggregation. It estimates a mean from random samples. It is useful for data scarcity.

For all the experiments, the algorithms were used with default parameters, and no optimisation procedure was performed. The testing process was carried out using 10-fold cross-validation and performing ten repetitions.

The results of the classification models are presented with the F-score and the Area Under the Curve (AUC) metrics. The accuracy of the regression models is presented in terms of the Spearman correlation coefficient (ρ), and the Mean Absolute Error (MAE). It is important to note that higher correlation coefficients and lower mean absolute error are desirable outcomes of the testing process.

Furthermore, we carried out paired samples t-tests to check statistical significance of the results of the feature sets. The symbol v means that the score is significantly higher than the baseline and the symbol * means the opposite, that the score is significantly lower than the baseline. No significance tests have been carried out on the performance of the different algorithms.

5.4.4 Feature Selection

Feature selection refers to the selection of attributes for a specific machine learning task. It aims to extract the best features using dimensionality reduction methods to improve the outcome of machine learning. We consider relevant to dedicate a section to feature selection as we have experienced a pattern when performing feature selection for the three different feature sets.

To determine the best-performing feature subsets for each quality predictor, we applied a filter-based feature selection (CfsSubsetEval attribute evaluator and the Best First search method) to our feature datasets (BASE, LING, BALI). Results are given independently of the classifier used, as the Best First algorithm looks for features that have a high correlation with the class and low correlation between each other.

6 Results

6.1 Results for the classification task

In this section, the results of the classification tasks are presented. We report the results for each quality indicator. The result tables show the F-score, usually used to evaluate machine learning models, and the AUC, together with their standard deviation in parentheses.

The two classes into which we will classify our dataset are the post-edit (PE) and the translate (T) class. Those sentences whose machine translation is of good quality and can be used for post-editing belong to the first class.

As we saw in Table 5, the distribution of our corpus into the two classes is unbalanced. The class T represents 9%. This makes the classification task difficult as the algorithms tend to classify all instances as PE and yet obtain outstanding results, as there are many more instances for PE. The 10-fold cross-validation divides the dataset into 458 training instances and 51 testing instances. In Table 8, we can see the confusion matrix for the averaged values of all tests.

classified as	**PE**	**T**
PE	45	5
T	1	0

Table 8: Confusion matrix for classification

The specificity of our models, the proportion of actual negatives that are correctly identified, is in average non-existent. Therefore, we shift our focus to evaluate the models, and our new goal is to investigate the identification of the true negatives, that is to say, we are interested in the ability of our models for classifying the T instances as such, in their specificity. For that reason, we provide the AUC (Area under the curve) metric, which takes into account the false positives and therefore represents better the performance of each model. This measure can be interpreted as the average value of specificity for all possible values of sensitivity. It ranges from 0.5 to 1, and a score above 0.70 is considered to be high.

6.1.1 Classification based on Quality

Overall, results are not good enough. Table 9 shows that the F-score measures are very high. The highest scores are achieved by the SMO and Bagging algorithms for all feature sets. The only significant difference across feature sets is that BALI offers poorer results with the L algorithm. However, the F-scores fail to capture the total performance of our classifier. The AUC results are more suited for our purposes.

F-score ↑	**BASE**	**LING**	**BALI**
L	0.96(0.01)	0.95(0.01)	0.94(0.02) *
MLP	0.95(0.01)	0.94(0.02)	0.94(0.02)
SMO	**0.96(0.00)**	**0.96(0.00)**	**0.96(0.00)**
IB1	0.94(0.02)	0.94(0.02)	0.93(0.02)
Bagging	**0.96(0.00)**	**0.96(0.00)**	**0.96(0.00)**

Table 9: F-score for classification based on Quality

As we can see in Table 10, AUC results are low, which probably describe the ability of the model to distinguish sentences for PE and T more accurately.. The highest AUC is achieved by the BALI feature set and the Bagging algorithm, closely followed by LING and MLP. However, these differences are not significant.

AUC ↑	**BASE**	**LING**	**BALI**
L	0.58(0.16)	0.56(0.18)	0.52(0.17)
MLP	0.53(0.17)	0.59(0.17)	0.55(0.16)
SMO	0.50(0.00)	0.50(0.00)	0.50(0.00)
IB1	0.54(0.10)	0.55(0.10)	0.53(0.08)
Bagging	0.55(0.15)	0.57(0.17)	**0.59(0.14)**

Table 10: AUC for classification based on Quality

Feature selection for classification based on Quality

Feature selection for classification based on Quality chooses two features for BASE, 5 for LING and eight for BALI. The latter contains five linguistic features and three baseline features.

#	BASE	LING	BALI
1	BSF2	short_np	short_np
2	BSF17	finite_v	finite_v
3		ellipticsubject_vs	ellipticsubject_vs
4		ovw	ovw
5		poly_words	poly_words
6			BSF2
7			BSF6
8			BSF17

Table 11: Feature selection for classification based on Quality

If we look at the F-Score in Table 12, it improves in some cases compared to the classification model without feature selection, but only by 1 or 2 points. Apart from SMO and Bagging, the L algorithm provides also the best results across all feature sets. However, now the BALI dataset does not provide significantly poorer results.

F-score ↑	**BASE**	**LING**	**BALI**
L	**0.96(0.00)**	**0.96(0.00)**	**0.96(0.00)**
MLP	0.96(0.01)	0.95(0.01)	0.95(0.01)
SMO	**0.96(0.00)**	**0.96(0.00)**	**0.96(0.00)**
IB1	0.95(0.01)	0.96(0.01)	0.94(0.02)
Bagging	**0.96(0.00)**	**0.96(0.00)**	**0.96(0.00)**

Table 12: F-score for classification based on Quality with feature selection

Regarding the AUC (See Table 13), the scores increase in most cases, but not greatly. LING and the L algorithm achieve the highest score, but it is not statistically significant. This measure is closely followed by Bagging and the BASE and BALI feature sets. The most significant improvement compared to Table 10 is achieved with the BASE feature set and the Bagging algorithm. SMO remains the same and experiences no variation. There is one significant difference, as the BALI feature set with IB1 is now significantly better than the baseline.

AUC ↑	BASE	LING	BALI
L	0.58(0.14)	**0.63(0.13)**	0.60(0.15)
MLP	0.60(0.15)	0.52(0.17)	0.50(0.16)
SMO	0.50(0.00)	0.50(0.00)	0.50(0.00)
IB1	0.44(0.13)	0.53(0.12)	0.56(0.12) v
Bagging	0.63(0.15)	0.61(0.15)	0.63(0.15)

Table 13: AUC for classification based on Quality with feature selection

6.1.2 Classification based on Time

Again, the F-scores for Time are very high in all cases, as they are all above 0.9 (See Table 14). With the BALI feature set, the results are significantly worse than the baseline with L and MLP. The scores are similar, but the best one is achieved by LING and SMO, although it is not significant. The ability of this model to identify true negatives, that is, the sentences to be translated, is doubtful, however.

F-score ↑	BASE	LING	BALI
L	0.95(0.01)	0.95(0.01)	0.94(0.02) *
MLP	0.95(0.01)	0.94(0.02)	0.93(0.03) *
SMO	0.95(0.01)	**0.95(0.00)**	0.95(0.01)
IB1	0.91(0.03)	0.91(0.02)	0.91(0.03)
Bagging	0.95(0.01)	0.95(0.01)	0.95(0.01)

Table 14: F-score for classification based on Time

Regarding AUC, we can see in Table 15 how the highest score is obtained with BALI and L, but overall the results are very low. There is one significant difference, for IB1, as the LING feature set provides a significantly better result than the baseline.

AUC ↑	BASE	LING	BALI
L	0.61(0.13)	0.59(0.13)	**0.64(0.13)**
MLP	0.56(0.13)	0.59(0.15)	0.55(0.15)
SMO	0.50(0.00)	0.50(0.00)	0.50(0.00)
IB1	0.49(0.10)	0.61(0.11) v	0.51(0.07)
Bagging	0.57(0.15)	0.63(0.12)	0.58(0.14)

Table 15: AUC for classification based on Time

Feature selection for classification based on Time

Table 16 shows the application of feature selection to the three feature sets. Feature selection selects the same feature for all, the length of the sentence in words (if we recall, BSF1 refers to the number of tokens in the source sentence). This single feature has such a weight that on its own it obtains good results.

#	BASE	LING	BALI
1	BSF1	sentence_len	sentence_len

Table 16: Feature selection for classification based on Quality

To further inspect this phenomenon, we performed a correlation analysis between the length of the sentence and the PE time. Their Spearman correlation is of **0.63**, which is a strong value. In Figure 6 we can observe the relationship between these two variables.

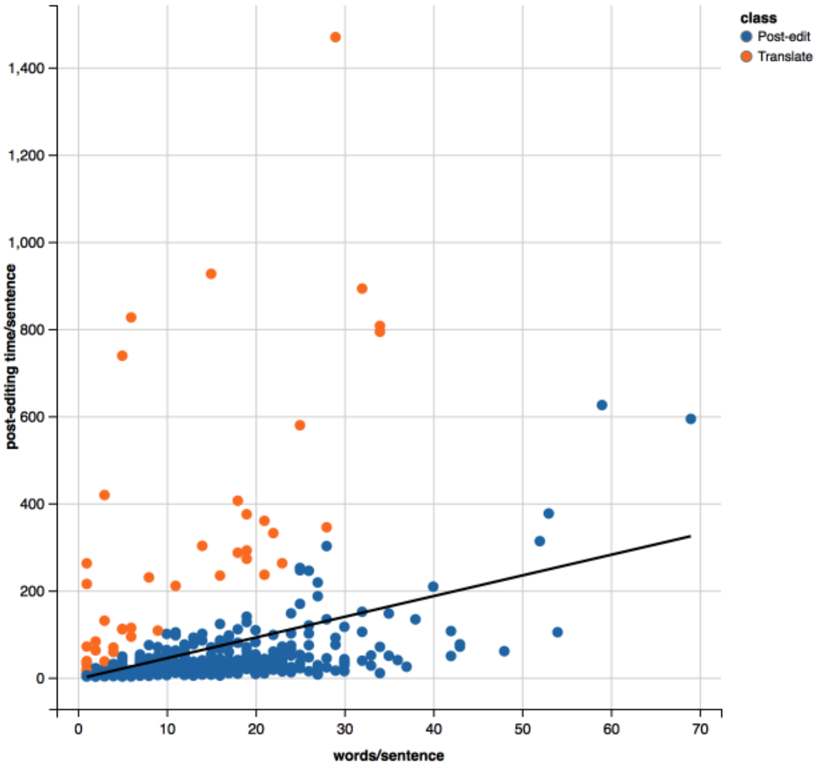

Figure 6: Correlation between sentence length and PE time

As all feature sets consist of the same feature, we only provide one result per algorithm.

Only with one feature, high F-scores are achieved, only 1 point below the highest achieved until now (See Table 17). All algorithms provide the same result except the IB1, which performs slightly worse.

F-score ↑	BASE/LING/BALI
LR	**0.95(0.00)**
MLP	**0.95(0.00)**
SMOreg	**0.95(0.00)**
IB1	0.95(0.01)
Bagging	**0.95(0.00)**

Table 17: F-score for classification based on Time with feature selection

Again, using only one feature, we obtain good results as seen in Table 18. In this case, IB1 that performs worse regarding F-score, here it provides the best score. This highest value falls only 1 point below the highest score achieved.

AUC ↑	BASE/LING/BALI
LR	0.57(0.18)
MLP	0.57(0.18)
SMOreg	0.50(0.00)
IB1	**0.63(0.13)**
Bagging	0.56(0.11)

Table 18: AUC for classification based on Time with feature selection

6.1.3 Classification based on HTER

In general, the F-scores obtained for the HTER classification are high as shown in Table 19. The best performing algorithm is SMO for all feature sets. The worst one is IB1. There is one significant difference as the BALI feature set performs worse than the baseline for the L algorithm.

F-score ↑	BASE	LING	BALI
L	0.94(0.01)	0.94(0.01)	0.92(0.02) *
MLP	0.94(0.01)	0.93(0.02)	0.93(0.02)
SMO	**0.95(0.00)**	**0.95(0.00)**	**0.95(0.00)**
IB1	0.91(0.03)	0.91(0.02)	0.92(0.02)
Bagging	0.94(0.01)	0.94(0.01)	0.94(0.01)

Table 19: F-score for classification based on HTER

In terms of AUC, the model obtains reasonable scores for HTER with a coefficient of 0.69. (See Table Table 20). The highest score is achieved by LING and Bagging, although not significantly different than the baseline. SMO provides the weakest results.

AUC ↑	BASE	LING	BALI
L	0.68(0.12)	0.60(0.12)	0.63(0.12)
MLP	0.68(0.12)	0.65(0.14)	0.63(0.14)
SMO	0.50(0.00)	0.50(0.00)	0.50(0.00)
IB1	0.53(0.08)	0.60(0.10)	0.55(0.07)
Bagging	0.62(0.13)	**0.69(0.12)**	0.66(0.11)

Table 20: AUC for classification based on HTER

Feature selection for classification based on HTER

In this case, the feature selection chooses 3 features for BASE, 4 for LING and 5 for BALI as we can see in Table 21. Of these last ones, 4 are linguistic features, and 1 is a baseline feature.

#	BASE	LING	BALI
1	BSF1	top_np	top_np
2	BSF2	adjp	adjp
3	BSF5	punct	punct
4		sentence_len	sentence_len
5			BSF2

Table 21: Feature selection for classification based on Quality

The highest F-score is the same as without feature selection, and it is obtained by the BASE feature set with three algorithms, MLP, SMO and Bagging (See Table Table 22).

There are no significant differences, BALI and L do not perform significantly worse anymore.

F-score ↑	**BASE**	**LING**	**BALI**
L	0.94(0.01)	0.94(0.01)	0.94(0.00)
MLP	**0.95(0.00)**	0.94(0.01)	0.94(0.00)
SMO	**0.95(0.00)**	0.94(0.00)	0.94(0.00)
IB1	0.93(0.02)	0.92(0.02)	0.93(0.01)
Bagging	**0.95(0.00)**	0.94(0.01)	0.94(0.01)

Table 22: F-score for classification based on HTER with feature selection

With HTER and feature selection the best results are achieved among all classifiers as shown in Table 23. BASE and L obtain the highest score. With this algorithm, BALI performs significantly worse.

AUC ↑	**BASE**	**LING**	**BALI**
L	**0.71(0.13)**	0.68(0.11)	0.56(0.12) *
MLP	0.69(0.12)	0.69(0.13)	0.66(0.13)
SMO	0.50(0.00)	0.50(0.00)	0.50(0.00)
IB1	0.61(0.13)	0.53(0.13)	0.60(0.11)
Bagging	0.68(0.11)	0.70(0.11)	0.69(0.1)

Table 23: AUC for classification based on HTER with feature selection

6.2 Results of the regression tasks

As we have seen, our models for classification obtain high F-scores but still fail to distinguish between the two classes PE and T. Since we have already defined thresholds for each quality indicator, another approach to the task is to perform a regression task and then apply the threshold on the predicted value. That is why we also built models for regression. Results are presented next.

In this section, the results of the regression tasks are presented. We report the results for each quality indicator (Quality, Time, HTER), using the three different feature sets. The result tables show the Spearman's correlation coefficient (ρ) and the MAE, together with their standard deviation in parentheses.

Spearman correlation is interpreted according to the following scale. Between 0.00-0.19 it is considered “very weak”, between 0.20-0.39 it is just “weak”, “moderate” if it falls between 0.40-0.59, “strong” between 0.60-0.79, and “very strong” if it is above 0.80.

6.2.2 Regression for Quality prediction

Overall, results are low as all correlations fall into the weak correlation category. If we look at the correlation coefficient in Table 24, the results show that the highest correlation coefficient is achieved by the SMOreg and the BALI feature set, and it is significant although it is a low score. The LING feature set achieves higher scores with 2 out of the 5 algorithms. However, these results are not significant.

ρ ↑	**BASE**	**LING**	**BALI**
LR	0.26(0.12)	0.25(0.13)	0.23(0.13)
MLP	0.16(0.15)	0.12(0.16)	0.10(0.16)
SMOreg	0.16(0.16)	0.25(0.15)	**0.27(0.15) v**
IB1	0.08(0.17)	0.14(0.15)	0.13(0.13)
Bagging	0.16(0.15)	0.24(0.14)	0.21(0.14)

Table 24: Spearman’s correlation for Quality prediction

Regarding the MAE, the algorithm that gives the best scores is again SMOreg for all feature sets (See Table 25). With the MLP algorithm, BALI and LING provide significantly worse results (higher MAE) while with LR and Bagging, the LING feature set provides better results than the baseline.

MAE ↓	**BASE**	**LING**	**BALI**
LR	0.48(0.05)	0.47(0.05)	0.48(0.05)
MLP	0.58(0.14)	0.71(0.14) v	0.68(0.15) v
SMOreg	**0.36(0.08)**	**0.36(0.08)**	**0.36(0.08)**
IB1	0.48(0.09)	0.48(0.08)	0.47(0.09)
Bagging	0.50(0.05)	0.48(0.05)	0.49(0.05)

Table 25: MAE for Quality prediction

Feature selection for Quality score prediction

The most remarkable thing to mention from Table 26 is that BALI seems to correlate more with solely linguistic features and disregards any BASE feature. As the BALI and LING feature sets have the same features, we present only the results for the BALI feature set. If we perform again the analysis, this time only with the selected features for each feature set, the results are the following:

#	**BASE**	**LING**	**BALI**
1	BSF1	long_vp	long_vp
2	BSF2	finite_v	finite_v
3	BSF11	dep_cl	dep_cl
4	BSF12	personal_pr	personal_pr
5	BSF16	oovw	oovw
6	BSF17	neg	neg
7		sym	sym
8		no_stop	no_stop

Table 26: Feature selection for Quality prediction

If we look at the correlation coefficient in Table 27, we can see how results are worse for BASE, but generally higher for BALI. In fact, BALI offers significantly higher results in 3 out of the 5 algorithms. With the BALI feature set and the Bagging algorithm, the highest correlation coefficient is obtained.

ρ ↑	**BASE**	**BALI**
LR	0.23(0.13)	0.26(0.13)
MLP	0.19(0.14)	0.24(0.13)
SMOreg	0.13(0.17)	0.26(0.13) v
IB1	-0.06(0.14)	0.14(0.16) v
Bagging	0.14(0.14)	**0.27(0.14) v**

Table 27: Spearman's correlation for Quality prediction with feature selection

For the MAE, results are similar. Now, BALI presents results significantly better (lower scores) with LR, IB1, and Bagging. Moreover, with feature selection, the algorithm MLP does not longer provide significantly worse results for BALI. SMOreg remains the same and offers the lowest MAE scores.

MAE ↓	BASE	BALI
LR	0.49(0.05)	0.48(0.05) *
MLP	0.54(0.14)	0.55(0.16)
SMOreg	**0.36(0.08)**	**0.36(0.08)**
IB1	0.53(0.09)	0.46(0.07) *
Bagging	0.50(0.05)	0.48(0.05) *

Table 28: MAE for Quality prediction with feature selection

6.1.2 Regression for time prediction

In contrast to the previous task, some high correlations are achieved with Time. The correlation scores shown in Table 29 are the highest obtained. If we look at the correlation coefficient, the results show that the highest correlation coefficient is achieved by the SMOreg and the BASE feature set, although similar results are achieved with the other two feature sets. The BALI and LING feature sets offer significantly poorer results for LR, MLP, and SMOreg.

ρ ↑	BASE	LING	BALI
LR	0.52(0.11)	0.41(0.10) *	0.35(0.13) *
MLP	0.41(0.12)	0.29 (0.14)	0.26(0.14) *
SMOreg	**0.61(0.09)**	0.60(0.09)	0.57(0.10)*
IB1	0.42(0.13)	0.33(0.13)	0.39(0.13)
Bagging	0.53(0.12)	0.52(0.11)	0.52(0.11)

Table 29: Spearman's correlation for Time prediction

Regarding the MAE, the algorithm that gives the best scores (that is, the lowest MAE) is again SMOreg for all feature sets (See Table 30). With LR, BALI and LING provide significantly worse results. There are no other significant differences.

MAE ↓	BASE	LING	BALI
LR	57100.43(14976.59)	64746.11(14886.58) v	62991.86(14181.10) v
MLP	80669.65(35091.43)	83396.77(26158.16)	91067.04(35896.71)
SMOreg	**43444.70(17416.57)**	43818.79(17161.02)	43992.62(17102.92)
IB1	62087.69(17245.52)	67049.02(17927.93)	68919.64(19433.73)
Bagging	57553.41(14848.39)	58034.42(15236.15	58653.11(15018.95)

Table 30: MAE for Time prediction

Feature selection for Time prediction

Again, the most remarkable fact from Table 31 is that the BALI feature set selects, as the most correlating features, 11 linguistic features.

#	BASE	LING	BALI
1	BSF1	long_np	long_np
2	BSF11	adjp	adjp
3		advp	advp
4		def_art	def_art
5		coor	coor
6		neg	neg
7		poly_words	poly_words
8		dom_words	dom_words
9		sym	sym
10		long_sentence	long_sentence
11		sentence_len	sentence_len
12			BSF16

Table 31: Feature selection for Time prediction

As we can see in Table 32, with feature selection, the results for the correlation coefficient improve with the baseline features (we must take into account that only two features have been selected) and with the other two features sets as well, the results are consistently higher. The highest score for Time prediction is obtained by all feature sets with the SMOreg algorithm and by BASE with LR and MLP. With these last two algorithms, LING and BALI offer significantly weaker results.

ρ ↑	**BASE**	**LING**	**BALI**
LR	**0.62(0.09)**	0.52(0.10)*	0.51(0.10)*
MLP	**0.62(0.09)**	0.44(0.15)*	0.44(0.15)*
SMOreg	**0.62(0.09)**	**0.62(0.09)**	**0.62(0.09)**
IB1	0.40(0.12)	0.29(0.11)*	0.31(0.13)
Bagging	0.53(0.10)	0.51(0.11)	0.52(0.10)

Table 32: Spearman's correlation for Time prediction with feature selection

Regarding the MAE, as for quality scores, the lowest MAE is achieved by the SMOreg algorithm for all feature sets (See Table 33). There are no significant differences.

MAE ↓	BASE	LING	BALI
LR	54786.71(14492.77)	57256.41(13379.31)	57654.70(13508.30)
MLP	73732.22(54788.18)	75826.96(30559.53)	85100.55(41936.21)
SMOreg	43321.61(17575.40)	43300.57(17050.35)	**43230.68(17121.45)**
IB1	61182.03(17288.68)	64404.09(19337.92)	61791.65(19540.86)
Bagging	57621.89(15138.54)	57150.58(15235.23)	57682.93(15284.62)

Table 33: MAE for Time prediction with feature selection

6.1.3 Regression for HTER prediction

For HTER prediction, the results are the worst (See Table 34). Regarding the correlation coefficient, the results show no significant differences. SMOreg provides the best results with the BALI feature set. BALI and LING offer higher results that the baseline with MLp, IB1, and Bagging. The correlations are consistently lower than for quality and time prediction.

ρ ↑	BASE	LING	BALI
LR	0.13(0.15)	0.06(0.15)	0.14(0.14)
MLP	0.02(0.16)	0.07(0.17)	0.10(0.15)
SMOreg	0.19(0.16)	0.18(0.13)	**0.20(0.14)**
IB1	0.10(0.15)	0.14(0.15)	0.17(0.15)
Bagging	0.04(0.13)	0.17(0.12)	0.06(0.13)

Table 34: Spearman's correlation for HTER prediction

Regarding the MAE, the algorithm that gives the best scores (that is, the lowest MAE) is again SMOreg for all feature sets as shown in Table 35. With MLP, provides significantly worse results.

MAE ↓	BASE	LING	BALI
LR	11.97(1.55)	12.42(1.60)	12.35(1.55)
MLP	14.35(3.97)	17.24(4.77)	18.04(4.43) v
SMOreg	9.70(2.10)	9.73(2.08)	**9.70(2.07)**
IB1	12.85(2.09)	12.75(2.02)	11.92(2.16)
Bagging	12.20(1.60)	11.88(1.41)	12.06(1.53)

Table 35: MAE for HTER prediction

Feature selection for HTER prediction

In this case, the BALI feature set selects five linguistic features and three baseline features as can be seen in Table 36.

#	BASE	LING	BALI
1	BSF4	personal_pr	oovw
2	BSF6	oovw	neg
3	BSF8	neg	poly_words
4	BSF12	poly_words	sym
5		sym	no_stop
6		no_stop	BSF8
7			BSF12
8			BSF17

Table 36: Feature selection for HTER prediction

We can see in Table 37 how feature selection improves the correlation coefficient for the BALI feature set. BALI gets a result significantly better than the baseline using MLP. However, the SMOreg algorithm provides the best results with the BASE feature set.

ρ ↑	BASE	LING	BALI
LR	0.15(0.13)	0.01(0.13)*	0.11(0.13)
MLP	-0.01(0.14)	0.08(0.14)	0.17(0.15) v
SMOreg	**0.21(0.16)**	0.14(0.13)	0.17(0.15)
IB1	0.07(0.13)	0.05(0.14)	0.08(0.13)
Bagging	0.05(0.14)	0.07(0.13)	0.11(0.14)

Table 37: Spearman's correlation for HTER prediction with feature selection

The MAE decreases in some cases, especially for the MLP algorithm. Now, LR and MLP do not present significantly poorer results for the BALI and LING feature sets. Moreover, with feature selection, the SMOreg algorithm offers the lowest MAE for all feature sets.

MAE ↓	BASE	LING	BALI
LR	12.00(1.50)	12.35(1.57)	12.22(1.54)
MLP	13.33(4.47)	14.46(4.20)	14.01(4.57)
SMOreg	**9.69(2.09)**	9.71(2.11)	9.73(2.12)
IB1	13.76(2.48)	12.47(1.67)	13.93(2.43)
Bagging	12.07(1.60)	12.13(1.48)	12.00(1.56)

Table 38: MAE for HTER prediction with feature selection

6.3 Discussion

We aimed to design a model to recommend whether a sentence should be post-edited or translated based on quality estimation of MT by providing three different feature sets and three different quality indicators. We approached the task of quality estimation as a supervised machine learning classification task. We tested our data on five different algorithms and presented their results in terms of F-score and AUC. Given the unreliable ability of the trained models to identify the sentences to be translated, we also tested a different approach where we predicted a post-editing effort indicator and then assigned this value to the binary class “post-edit” or “translate”. Again, we used the five different algorithms mentioned above to train regression model and presented the results in terms of Spearman correlation and MAE. In the discussion that follows, we analyse the results found during the experimentation.

For classification, the highest F-score is 0.96 and it is achieved by the different algorithms such as L, SMO, and Bagging, using quality score as the quality indicator. However, we have seen how F-scores may not describe fully the performance of our models, as we are interested in their specificity. That is why we also provide the AUC measure. The highest value of AUC is obtained with HTER and feature selection using a logistic regression. The worst performing algorithm for AUC is SMO, that is consistently 0.50 which equates to chance. We acknowledge that high scores do not translate to optimal performance.

Regarding the regression task, in general, we can observe low to high correlations. The highest score is 0.62, which is a high score, and is achieved by the SMOreg algorithm using Time as the quality indicator and all feature sets.

We take the chance to explore the findings at different levels. First, we contrast the differences between the algorithms used. Second, we compare the linguistic and baseline features and their relevance for classification and regression tasks with and without feature selection. Third, we analyse the quality indicators used in this study. Fourth, we discuss how the corpus used for this task might not be suitable. Finally, we identify the limitations of our work.

Algorithms

We must take into account that we have not performed optimization for any of the algorithms and we carried out the tasks using the default parameters, except for SMO, for which we opted for a non-linear kernel (RBF). Furthermore, we did not carry out significance tests to compare the performance of each algorithm, that is why we report the results for algorithms that work best across all feature sets.

For classification, the performance of the algorithms is different depending on whether the F-score or the AUC is taken into consideration. For F-score, the implementation of SMO for classification seems to provide the best results across feature sets. However, this algorithm is the one that performs worse in terms of AUC, as it seems it is the algorithm that fails more to distinguish between the two classes. The best performing algorithm both in terms of F-score and AUC is the logistic regression (L). Nonetheless, L does not perform so well with BALI and sometimes offers significantly poorer results.

For regression, results show that the implementation of SMO offers the highest correlation scores across feature sets for quality score and time prediction. The second best performing algorithm is Bagging, however this algorithm is known to be very dependent on the training data. We can observe how LR and MLP work best for the BASE feature set regardless of the quality indicator. IB1 and Bagging work specially well with the LING feature set.

If we look at MAE, the algorithm that consistently offers the lowest scores and therefore, the best results is SMOreg in all cases.

Feature sets

Results show that there are not many significant differences among the three feature sets. F-scores are similar for all feature sets. If we look at the AUC, in general the LING and BALI provide better results, offering the highest values for Quality score and Time.

In the regression task, the BASE feature set achieves the highest scores in most of the cases, although the BALI and LING feature sets also can get the higher scores and provide significantly better results than the baseline for Quality prediction. In terms of MAE, the baseline offers the lowest results in most cases as compared to BALI and LING.

The fact that there are no significant differences and that BALI and LING are able to outperform the baseline under specific conditions is an interesting outcome of our study. This means feature sets perform similarly and their optimal combination should be explored.

Nonetheless, we can see how both tasks failed to achieve the expected results. This tells us that our three feature sets are not suited for these tasks and are not discriminative enough as to provide acceptable results for this corpus. This leads us to consider that the features are not informative enough for what we aim to predict.

Feature selection

It is worth to dedicate a section to study the effect that feature selection has on the results. In all, feature selection offers higher scores for each quality indicator, except for the F-scores of our classifiers, that remain constant throughout the whole task. First, we will revise the two unique feature sets (BASE and LING) individually and then we will observe the BALI feature set.

The features that correlate more with each of the three quality indicators for the classification task in the BASE feature set are BSF1 and BSF2, that are selected in two out of the three tasks. Other selected features are BSF5, BSF6, BSF17. These are respectively, “LM probability of target sentence”, “number of occurrences of the target word within the target hypothesis (averaged for all words in the hypothesis - type/token ratio)”, and “number of punctuation marks in the target sentence”. These features rely on the target text, that is, the machine translation. It seems that the availability of the MT output, and not of source text characteristics, is relevant for classification. Nonetheless, in the regression task, the features that are selected in every task are BSF1 and BSF11. These are the “number of tokens in the source sentence” and the “percentage of unigrams in quartile 4 of frequency (higher frequency words) in a corpus of the source language”. It is worth mentioning that both rely on the source text and not on the machine translation output. Our study also relies on the sole use of the source text for quality estimation.

Regarding the LING feature set, classification selects a different feature subset for each quality indicator (See Section 5.3.1 for an explanation of each feature). For Quality, the following features are selected: short_np, finite_v, ellipticsubj_vs, oovw, poly_words. For HTER, these are selected: top_np, adjp, punct, and sentence_len. Surprisingly, the latter is the only one selected for Time. This means that sentence length has a strong correlation with Time.

In the case of regression, the features that get selected in all three tasks are: neg and sym. Other selected features in two out of the three prediction tasks are personal_pr, oovw, dom_words and no_stop. Other linguistic features selected are the following: long_np, adjp, advp, def_art, finite_v, long_vp, dep_cl, coor, poly_words, long_sentence, sentence_len.

In we look at both tasks, the use of out of vocabulary words, polysemic words and the length of the sentence seem to have an important weight for quality estimation. Furthermore, PoS also plays an important role, as nouns, verbs, and adjectives are considered for feature selection. Semantics and syntax are also important as ellipticsubj_vs, and poly_words are selected. In total, 19 out of the 28 linguistic features are chosen at least once.

Last, if we look at the results of feature selection for the BALI feature set, we see that there is a majority of linguistic features for classification based on Quality and HTER over baseline features. We also want to emphasize that for Time, very good results are obtained only with one feature: the length of the sentence. It seems to be a good indicator of quality. In the case of the regression task, we can observe how almost all selected features for Quality and Time prediction belong to the LING feature set. In the case of HTER prediction, five out of the eight selected features are from LING too.

In general, when BASE and LING are combined, the linguistic features tend to be selected over baseline features, as these seem to fail to represent our quality indicators. These results are hopeful and open a path in the field as they show the relevance of linguistic features.

Quality indicators

We can also analyse the results depending on the quality indicator, although we must take into account each quality indicator was trained on different data, provided by each threshold. For classification, in terms of F-score, the highest result is **0.96** and it is achieved by the indicator Quality and L, SMO, and Bagging algorithms across all feature sets. In terms of AUC, the highest result is **0.71** and it is obtained with HTER as the quality indicator and the baseline feature set with logistic regression. Coincidentally, when applying the threshold of HTER, we get the highest number of sentences of class T, as there are 56 sentences (See Table 5). The fact that classification based on HTER is the one that has more T samples may be influencing the results.

In the case of regression, the highest correlation score is achieved by the indicator Time at **0.62** with all feature sets**,** which is a high correlation. Time is the less subjective quality indicator. It refers to the time spent in the task of PE for each machine translated sentence and thus, it is collected during the whole process of annotation. In contrast, HTER is only computed afterwards, once all changes have been made.

Let's imagine a difficult sentence to post-edit and a translator that reads the sentence carefully and after a while, decides what changes to make. The reading of the sentence and the thinking of how to improve it is recorded in the time. Nonetheless, this cannot be conveyed with an a posteriori HTER measure. Being time the most objective measure for PE effort, it is coincidentally the most suited for prediction.

We can also understand why Quality is the indicator that offers the poorest results. The annotation of quality scores was performed by one sole translator and therefore it is the most subjective indicator. Furthermore, there are issues associated with how each scorer interprets the description of the quality scale, leading to a larger variability of scores for each sentence, and thus poorer performance. This seems to be a common issue within the quality estimation literature which is often ameliorated by having more scores per sentence and then taking the average of scores (See Lacruz, Denkowski, & Lavie, 2014).

Corpus

We would like to emphasize some key characteristics of our corpus that may also have impact on the results. We acknowledge that the corpus used in this thesis is not big as compared to other studies on QE. Furthermore, the performance of the MT system was excellent, as 90% of the sentences were considered useful for PE. This unbalanced distribution translates into data scarcity, which led to difficulties in performing the machine learning tasks, probably aggravated by the low informativeness of the features to predict the specific indicators we set to predict

Our corpus was annotated in its entirety (not only when the features were the cause of errors). This results in both sentences to post-edit and sentence to translate to have features annotated and our models could not distinguish them. That is to say, if most machine translations provided by a system are good or, in other words, belong to a specific class, the features carry less weight and the whole dataset is less informative for QE. Therefore, we argue, the better translations a MT system provides, the more difficult it is to build a model for QE that relies on generic system independent or linguistic features.

Moreover, we also argue that QE for domain-specific texts is a hard task. Domain-adapted MT systems provide either very good or very bad translations, as the source sentence can either belong or not to the domain. This means that the quality scale is not as spread as that of a general purpose MT engine and tends to have a distribution inclined to the opposite ends of the scale.

Possible limitations

In all, this approach for quality estimation cannot be implemented into production unless further research is conducted because the models are not reliable enough as they are. We believe that the main improvements will come from two distinct approaches.

First, we also inspected the sentences that belong to the T class to learn more about why our classifier was not performing as desired. In general, they are very long or very short sentences (one-word sentences). You can see some examples together with their MT and their post-edition in Table 39.

Source	MT	Post-edition
© Copyright Orona	*&amp; # 169 ; Copyright Orona*	*© Copyright Orona*
TAPARA LOS AGUJEROS ED MONTAJE DE GUÍAS EN LA LOSA.	*TAPARA THE ED HOLES IN THE GUIDE SLAB ASSEMBLY.*	*COVER THE GUIDE ASSAMBLE HOLES IN THE SLAB*
Llave fija/tubo/carraca	*Fixed spanner/box/ratchet wrench*	*Fixed wrench/tube/ratchet.*
Instalando un deflector que forme una superficie inclinada de un mínimo de 45º con la horizontal, que al aplicarle una fuerza de 300N distribuida uniformemente en una superficie de 5cm2 de sección redonda o cuadrada, aplicada en ángulo recto y contra la pared en cualquier punto de una u otra cara, debe resistir:	*Installing a deflector that forms a minimum inclined surface of a minimum of 45 &amp; # 186; with the horizontal that by applying a force of 300N distributed evenly within a surface of 5cm2 of round or side it must resist:*	*Installing a deflector that forms a minimum inclined surface of a minimum of 45º to the horizontal that by applying a force of 300N distributed evenly within a surface of 5cm2 of round or square section applied the right angle and against the wall in any point of one or other face must resist:*

Table 39: Example sentences belonging to the T class

They do contain the identified linguistic features. Among all of them, we see how in these sentences words of the domain, abbreviations, the se particle, symbols, dependent clauses, noun phrases, etc. appear. However, these linguistic features are not found exclusively in the sentences that are badly translated, they are also encountered in many other sentences that the MT system managed to translate perfectly. That is why research using more discriminative linguistic features is necessary.

Furthermore, we believe that, combined with the features that did not manage to discriminate the classes well, the unbalanced classes might have had a significant impact on the poor results. In order to get a first glance at the performance of our classifiers with balanced classes, we chose the three best performing models for Quality, Time and HTER and run a quick test where we balanced their classes by applying the filter ClassBalancer in Weka. This converted the data into 254 samples of each class.

In Table 40 we can see the results achieved, more faithful as now not everything is classified as PE. The F-scores are much lower but the AUC is kept. It is worth to explore in the future the task of QE for a recommender system with balanced classes.

Quality indicator	Feature set	Algorithm	F-score	AUC
Quality	LING	L	0.572	0.623
Time	BALI	L	0.611	0.635
HTER	BASE	L	0.649	0.701

Table 40: Classification trial with balanced classes

Other factors that may have influenced negatively our results include the data sparsity and reliability of annotations. The use of a small data set, 509 sentences, may have played a role in the task as machine learning algorithms are known to need a lot of data to work efficiently. These can also be inferred from the results provided by the baseline, which usually performs very well and, in this case, the highest correlation score for the regression task is of 0.62. This data scarcity is also encountered in the quality ratings performed by the translators. Each sentence was only rated by one person, which may fail to provide an unbiased measure of its quality.

Another issue to take into account is the human factor in the annotation and PE process. The annotation of linguistic features was carried out by the author of this thesis, which is subject to human errors and inconsistencies. Furthermore, the PE task was carried out by means of an online platform which would not allow the supervision of the translators, so we cannot ensure the correct development of the task.

With regards to the feature set, the extraction of the 17 baseline features was performed using language models of a generic corpus used in WMT shared task. The very specific domain of our data set is not represented in it, which could also significantly harm the accuracy of the features.

7 Conclusions and Future Work

In this work we presented an approach to the task of quality estimation for MT based on a real set-up. The basis for our QE model involves the use of manually annotated linguistic features extracted from the source text to build a recommender system that classifies sentences for post-editing (PE) or for translation. Our results allow us to draw concluding remarks at different levels.

Despite not having definite results ready for implementation given that our models are not able to distinguish between the two classes of PE and T, we can still draw conclusions about the relevance of source text characteristics and the existing relationship among the three chosen quality indicators. Additionally, this work might prove relevant for any company considering the option of integrating QE for MT, as it is reports on the different steps to take, the issues found along the way and possible alternatives.

What is most important for this study is that it is a proof of concept and demonstrates the relevance of linguistic features. The feature set created in this thesis from scratch achieves and surpasses occasionally the baseline currently used for quality estimation. Sometimes these are overruled by the combination of both feature sets. Furthermore, feature selection chooses linguistic features over baseline features in most of the cases. However, we have seen that our features are not discriminative enough to classify the sentences for PE or translation.

Another major contribution of this study is the investigation of PE effort in its three dimensions and their translation into concrete measures, the three different quality indicators (Quality, Time and HTER). The correlation analysis performed confirms the existing relation among them and lets us get the complete picture of the work of the translators. Plus, we studied how good each of the quality indicators was for prediction and identified why Time may be the best quality indicator.

This study follows the work initiated by Felice & Specia (2012) concerning the use of linguistic features for quality estimation. As we have already mentioned, our results are not high in most cases and we identified the reasons why this may be so. These results are directly related with the future work that is to be done regarding this issue.

It is necessary to study the use of linguistic features in QE by means of a bigger dataset. The fact that even the baseline provided poor results is an indication of the low efficiency of the dataset. In terms of quality, an annotation effort involving more than one annotator per sentences could be organised so that the quality score would be more consistent. This would avoid the introduction of bias in the results. Plus, the combination of linguistic features together with baseline features should be explored.

Also, the identification of new features, more discriminative, should take place. Having a bigger dataset, there would be less data scarcity and hopefully the classes would be more balanced. This would allow looking for features present only in one class that would make the distinction easier.

Further work may include the investigation on how the quality indicators may be combined to provide a more informative label representing PE effort. Here we presented them separately. However, a combination of the three would be a very good representation of PE effort in its three dimensions.

Finally, the most logical step to follow this thesis is to develop a program to automatically annotate linguistic features. This can be done by means of Natural Language Processing toolkits, such as NLTK[1]. Two of the linguistic features were indeed annotated in this way as it was easier and faster and carried little room for mistakes. These are the dom_words and poly_words. You can see the code in Appendix C.

Annotating the linguistic features automatically would allow using much more data and making the process much faster, as manual annotation is time-consuming. Moreover, the result would be a more consistent annotation, as opposed to the work performed by a human.

[1] https://www.nltk.org/

On a personal level, I have learned the research and development process from the inception to the execution and testing of a proof of concept enclosed in a real setting. Particularly, I have learned about the essential importance of data collection, analysis, preparation, and processing for machine learning, as well as the different models and metrics to assess them. To post-edit or to translate… That was the question and it still is, as further research is needed to provide a definitive answer to this key question.

8 Appendices

A. Example of an annotated sentence

Para los casos en los que haya que sustituir las máquinas M33 originales (con brazos de freno de tambor), es necesario sustituir además de la máquina, la nueva armadura de la máquina que se envía.

long_np	short_np	top_np	adjp	pp	top_pp	advp
2	6	4	3	5	7	1

long_vp	nonfinite_v	finite_v	dep_cl	ellipsubj_vs	se_particle	personal_pr
2	2	3	3	1	1	1

def_art	coor	num_seq	abbrev	oovw	neg	poly_words
6	0	1	1	1	0	8

dom_words	sym	punct	no_verb	no_stop	long_sen	sen_len
2	0	4	0	0	1	35

B. Guidelines for the post-editing task

Introduction

Dear participant,

Thank you for helping enrich my thesis by providing your knowledge and expertise. This thesis intends to design a system of Quality Estimation of Machine Translation for MondragonLingua and their client Orona. To be able to build it, I need someone to post-edit sentences and to give an indication of their quality. These need to be provided by human translators, and that is when you come into play. You will perform two tasks:

- A practice task, in which you will postedit 5 sentences so that you familiarize with the environment.
- A real task, results of which will be used in my thesis.

Thank you very much for your collaboration.

Best regards,

Ona.

Methodology

You will be presented machine translated (MT) sentences in English, together with their source sentence in Spanish. You need to modify (postedit) the machine translated sentence as little as possible so that it bares the same meaning as its source. This postedition may include substitutions, deletions, insertions or reorderings. For each postedition, three things will be recorded:

- Your postedited version of the sentence
- A number provided by you indicating the quality of the machine translated sentence
- The time you take to postedit (this will be recorded automatically)

Here you can find an example of the task you'll need to perform:

- Source: Es necesario tapar esos agujeros ya que las patas traseras de la armadura apoyan sobre ellos.
- MT:These holes have to be covered as the frame rear feet support them.
- Postedited version: These holes need to be covered as the frame back legs support on them.
- Quality: 1 2 3 4 5

The tool: Matecat

The platform we will be using is called Matecat. Matecat is an open source online CAT tool. If you want more information, check this link: https://www.matecat.com/about/

To perform this task, you will be given two links, one for each task (practice task and real task). In one, you will have 5 sentences to familiarize yourself with the environment. In the other, you will have between 130 and 230 sentences to postedit, around 2180 words. When opening the link, you will be directed to the environment in which you will work. This is the following:

The platform shows the source sentence on the left side and its machine translation on the right side. This machine translated proposal is taken from a translation memory that contains all segments translated automatically. You need to post-edit the Machine Translate sentence in the right window taking into account the source sentence shown on the left window.

Quality rating

To rate the quality of the Machine Translation, please follow the scale proposed by Lacruz, Denkowski & Lavie (2014):

Rating	Criterion
1	Gibberish – The translation is totally incomprehensible
2	Non-usable – The translation has so many errors that it would clearly be faster to translate from scratch
3	Neural – The translation has enough errors that it is unclear if it would be faster to edit or translate from scratch
4	Usable – The translation has some errors but is still useful for editing
5	Very good – The translation is correct or almost correct

After post-editing each sentence, you need to add a **number** at the end of each sentence indicating the quality of the machine translated sentence following the aforementioned scale (1-5). Then, hit the button **TRANSLATED** to finish.

Please keep in mind that you need to write this number for each sentence.

If you feel that a certain machine translated sentence is **perfect** and needs no post-editing, just add a **number** from 1 to 5 at the end and hit the **TRANSLATED** button directly.

C. Post-editing survey

Dear participant,

Once you have finished the post-editing task, I would really appreciate it if you could answer a few questions about your background as well as your impressions on post-editing and the task itself.

Background

- What year were you born?
- What is your gender?
- Do you have formal studies in translation? If so, which?
- Are you a translator full-time?
- What are your fields of specialization? (Technology, law, medicine, economy, IT, media, literature, other…)
- How long have you been working in the translation industry?
- Which language pairs do you usually work with?
- Feel free to add any comments you like about your experience with the translation industry.

Post-editing and Machine Translation

- Had you heard about post-editing, before carrying out this task?
- Had you post-edited, before carrying out this task? If so, could you give a brief explanation?
- What is your attitude towards machine translation (and post-editing)? (positive, rather positive, rather negative, negative)

The post-editing task

- In general, after having performed the post-editing task, do you think it is helpful to post-edit previously machine translated sentences as opposed to translate from scratch?
- Feel free to add any comments you like.

D. Automatic annotation of linguistic features

dom_words

```
#Domain-specific vocabulary
#This program returns the number of domain-specific words in a sentence as contrasted against a list of domain words provided by the LSP MondragonLingua
import re, string
#open file with list containing domain-specific words
domain_vocab_list = open("domain_words.txt","r").readlines()
domain_vocab_words=[]
for line in domain_vocab_list:
        domain_vocab_words.append(line.strip("\n"))

#open file with list containing the 511 sentences
sentences = open("sentences.txt").readlines()
for line in sentences:
#removes punctuation
    exclude = set(string.punctuation)
    line = ''.join(ch for ch in line if ch not in exclude)
    domain_vocab=0
    sentence=line.split()
    for each_word in sentence:
        for each_vocab in domain_vocab_words:
            if each_word.lower() == each_vocab:
                domain_vocab+=1
    print (line, domain_vocab)
```

poly_words

```
#This program returns the number of polysemic words contained in a sentence. Polysemic words are those that have more than one sense.
import string
from nltk.corpus import wordnet as wn

#open file with list containing the 511 sentences
sentences = open("sentences.txt","r").readlines()

for line in sentences:
#removes punctuation
    exclude = set(string.punctuation)
    line = ''.join(ch for ch in line if ch not in exclude)
    sentence=line.split()
    polisem_words=0
    polisemic_words_list=[]
    for each_word in sentence:
        if ((len(wn.synsets(each_word.lower(),lang='spa'))) > 1): #if the synset is bigger than 1
               polisem_words+=1
               polisemic_words_list.append(each_word)
    print (line,"\t",polisem_words,"\t",polisemic_words_list)
```

9 References

Alegria I, de Ilarraza A, Labaka G, Lersundi M, Mayor A, Sarasola K. (2007). Transfer-based MT from Spanish into Basque: reusability, standardization and open source. In: *Lecture notes in computer science, vol 4394*. Springer, Heidelberg, 374–384.

Aranberri, N., & Pascual, J. A. (2018). Towards a post-editing recommendation system for Spanish–Basque machine translation, 21–30.

Arenas, A. G. (2008). Productivity and quality in the post-editing of outputs from translation memories and machine translation. *Localisation Focus*, *7*(1), 11-21.

Artetxe, M., Labaka, G., & Agirre, E. (2017). Learning bilingual word embeddings with (almost) no bilingual data. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (Vol. 1, pp. 451-462).

Artetxe, M., Labaka, G., & Agirre, E. (2018). A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. *arXiv preprint arXiv:1805.06297.*

Artetxe, M., Labaka, G., Agirre, E. & Cho, K. (2018). Unsupervised Neural Machine Translation. In *Proceedings of the Sixth International Conference on Learning Representations (ICLR 2018)*.

Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473.*

Bernth, A. (1999). A confidence index for machine translation. *Proceedings of TMI, 99*.

Bernth, A. & Gdaniec, C. (2001). MTranslatability. *Machine Translation. 16*, 175-218.

Bojar, O., Chatterjee, R., Federmann, C., Graham, Y., Haddow, B., Huang, S., ... & Monz, C. (2017). Findings of the 2017 conference on machine translation (wmt17). In *Proceedings of the Second Conference on Machine Translation*, 169-214.

Bowker, L., & Fisher, D. (2010). Computer-aided translation. *Handbook of translation studies*, *1*, 60-65.

Cho, K., Van Merriënboer, B., Bahdanau, D., & Bengio, Y. (2014). On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259.*

Dahlmann, L., Matusov, E., Petrushkov, P., & Khadivi, S. (2017). Neural machine translation leveraging phrase-based models in a hybrid search. *arXiv preprint arXiv:1708.03271.*

de Souza, J. G. C., Negri, M., Ricci, E., & Turchi, M. (2015). Online Multitask Learning for Machine Translation Quality Estimation. *In ACL (1)*, 219-228.

Dhariya, O., Malviya, S., & Tiwary, U. S. (2017, January). A hybrid approach for Hindi-English machine translation. In *Information Networking (ICOIN), 2017 International Conference*, 389-394. IEEE.

Dyer, C., Weese, J., Setiawan, H., Lopez, A., Ture, F., Eidelman, V., ... & Resnik, P. (2010, July). cdec: A decoder, alignment, and learning framework for finite-state and context-free translation models. In *Proceedings of the ACL 2010 System Demonstrations* (pp. 7-12). Association for Computational Linguistics.

Federico, M., Bertoldi, N., Cettolo, M., Negri, M., Turchi, M., Trombetti, M., ... & Massidda, A. (2014). The matecat tool. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: System Demonstrations* (pp. 129-132).

Federico, M., Cattelan, A., & Trombetti, M. (2012, October). Measuring user productivity in machine translation enhanced computer assisted translation. In *Proceedings of the Tenth Conference of the Association for Machine Translation in the Americas (AMTA)* (pp. 44-56). Madison, WI: AMTA.

Felice, M., & Specia, L. (2012, June). Linguistic features for quality estimation. In *Proceedings of the Seventh Workshop on Statistical Machine Translation* (pp. 96-103). Association for Computational Linguistics.

Forcada, M. L., Ginestí-Rosell, M., Nordfalk, J., O'Regan, J., Ortiz-Rojas, S., Pérez-Ortiz, J. A., ... & Tyers, F. M. (2011). Apertium: a free/open-source platform for rule-based machine translation. *Machine translation, 25(2)*, 127-144.

Fujita, A., & Sumita, E. (2017). Japanese to English/Chinese/Korean Datasets for Translation Quality Estimation and Automatic Post-Editing. In *Proceedings of the 4th Workshop on Asian Translation (WAT2017)*, 79-88.

Gdaniec, C. (1994, October). The Logos translatability index. In *Proceedings of the First Conference of the Association for Machine Translation in the Americas*, 97-105.

Graham, Y. (2015). Improving evaluation of machine translation quality estimation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 1804-1813.

Green, S., Heer, J., & Manning, C. D. (2013, April). The efficacy of human post-editing for language translation. In *Proceedings of the SIGCHI conference on human factors in computing systems* (pp. 439-448). ACM.

Hardmeier, C. (2011). Improving machine translation quality prediction with syntactic tree kernels. In *EAMT 2011, Leuven, Belgium, May 30, 2011-May 31, 2011* (pp. 233-240). European Association for Machine Translation (EAMT).

Hokamp, C. (2017). Ensembling Factored Neural Machine Translation Models for Automatic Post-Editing and Quality Estimation. *arXiv preprint arXiv:1706.05083.*

ISO/TC27. (2017). ISO 18587:2017 translation services: Post-editing of machine translation output: Requirements.

Joscelyne, A., van der Meer, J., Samiotou, A., & Ruopp, A. (2017). TAUS Machine Translation Report 2017.

Klein, G., Kim, Y., Deng, Y., Senellart, J., & Rush, A. M. (2017). OpenNMT: Open-Source Toolkit for Neural Machine Translation. *arXiv preprint arXiv:1701.02810*.

Koehn, P. (2009). Statistical machine translation. *Cambridge University Press.*

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., ... & Dyer, C. (2007, June). Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions* (pp. 177-180). Association for Computational Linguistics.

Koponen, M., Aziz, W., Ramos, L., & Specia, L. (2012, October). Post-editing time as a measure of cognitive effort. In *AMTA 2012 Workshop on Post-Editing Technology and Practice (WPTP 2012)* (pp. 11-20).

Krings, H. P. (2001). *Repairing texts: empirical investigations of machine translation post-editing processes* (Vol. 5). Kent State University Press.

Lacruz, I., Denkowski, M., & Lavie, A. (2014). Cognitive demand and cognitive effort in post-editing. *AMTA.*

Logacheva, V., Hokamp, C., & Specia, L. (2016, May). MARMOT: A Toolkit for Translation Quality Estimation at the Word Level. In *LREC.*

Martins, A. F., Junczys-Dowmunt, M., Kepler, F. N., Astudillo, R., Hokamp, C., & Grundkiewicz, R. (2017). Pushing the limits of translation quality estimation. *Transactions of the Association for Computational Linguistics*, *5*, 205-218.

Miyata, R., Hartley, A., Kageura, K., & Paris, C. (2017). Evaluating the Usability of a Controlled Language Authoring Assistant. *The Prague Bulletin of Mathematical Linguistics, 108(1)*, 147-158.

Moreau, E., & Vogel, C. (2014, August). Limitations of MT Quality Estimation Supervised Systems: The Tails Prediction Problem. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics* (pp. 2205-2216). Dublin City University and Association for Computational Linguistics.

O'Brien, S. (2004). Machine translatability and post-editing effort: How do they relate. *Translating and the Computer, 26.*

O'Brien, S. (2005). Methodologies for measuring the correlations between post-editing effort and machine text translatability. *Machine Translation 19(1)*, 37-58.

O'Brien, S. (2011). Towards predicting post-editing productivity. *Machine translation*, *25*(3), 197.

Plitt, M., & Masselot, F. (2010). A productivity test of statistical machine translation post-editing in a typical localisation context. *The Prague bulletin of mathematical linguistics*, *93*, 7-16.

Sawaf, H., Shihadah, M., & Yaghi, M. (2017). *U.S. Patent No. 9,798,720*. Washington, DC: U.S. Patent and Trademark Office.

Scott B, Barreiro A (2009) Openlogos MT and the SAL representation language. In: *Proceedings of the first international workshop on free/open-source rule-based machine translation*, 19–26

Sennrich, R., Firat, O., Cho, K., Birch, A., Haddow, B., Hitschler, J., ... & Nădejde, M. (2017). Nematus: a toolkit for neural machine translation. *arXiv preprint arXiv:1703.04357*.

Shah, K., Avramidis, E., Biçici, E., & Specia, L. (2013). QuEst–design, implementation and extensions of a framework for machine translation quality estimation. *The Prague Bulletin of Mathematical Linguistics*, *100*, 19-30.

Shah, K., Cohn, T., & Specia, L. (2015). A bayesian non-linear method for feature selection in machine translation quality estimation. *Machine Translation*, *29*(2), 101-125.

Snover, M., Dorr, B., Schwartz, R., Micciulla, L., & Makhoul, J. (2006, August). A study of translation edit rate with targeted human annotation. In *Proceedings of association for machine translation in the Americas* (Vol. 200, No. 6).

Snover, M., Madnani, N., Dorr, B. J., & Schwartz, R. (2009, March). Fluency, adequacy, or HTER?: exploring different human judgments with a tunable MT metric. In P*roceedings of the Fourth Workshop on Statistical Machine Translation* (pp. 259-268). Association for Computational Linguistics.

Søgaard, A., Ruder, S., & Vulić, I. (2018). On the Limitations of Unsupervised Bilingual Dictionary Induction. *arXiv preprint arXiv:1805.03620.*

Somers, H. (1999). Example-based machine translation. *Machine Translation, 14(2)*, 113-157.

Specia, L., (University of Sheffield, UK): Software and Tools for High Quality Translation -- Quality Estimation. META-FORUM 2013. September 19/20, 2013. Berlin, Germany.

Specia, L., Hajlaoui, N., Hallett, C., & Aziz, W. (2011, September). Predicting machine translation adequacy. In *Machine Translation Summit* (Vol. 13, No. 2011, pp. 19-23).

Specia, L., Paetzold, G. H., & Scarton C. (2015). Multi-level Translation Quality Prediction with QuEst++. *ACL-IJCNLP 2015 System Demonstrations*, Beijing, China. 115–120

Specia, L., Raj, D., & Turchi, M. (2010). Machine translation evaluation versus quality estimation. *Machine translation, 24(1)*, 39-50.

Tatsumi, M., & Roturier, J. (2010, November). Source text characteristics and technical and temporal post-editing effort: what is their relationship. In *Proceedings of the Second Joint EM+/CNGL Workshop Bringing MT to the User: Research on Integrating MT in the Translation Industry (JEC 10),* 43-51.

Uchimoto, K., Hayashida, N., Ishida, T., & Isahara, H. (2005). Automatic rating of machine translatability. *10th Machine Translation Summit (MT Summit X),* 235-242.

Underwood, N., & Jongejan, B. (2001, September). Translatability checker: A tool to help decide whether to use MT. In *Proceedings of MT Summit VIII: Machine Translation in the Information Age*, 363-368.

Vieira, L. N. (2014). Indices of cognitive effort in machine translation post-editing. *Machine translation*, *28*(3-4), 187-216.