Automatic Error Detection and Correction in Neural Machine Translation

A comparative study of Swedish to English and Greek to English

Anthi Papadopoulou
Abstract

Automatic detection and automatic correction of machine translation output are important steps to ensure an optimal quality of the final output. In this work, we compared the output of neural machine translation of two different language pairs, Swedish to English and Greek to English. This comparison was made using common machine translation metrics (BLEU, METEOR, TER) and syntax-related ones (POSBLEU, WPF, WER on POS classes). It was found that neither common metrics nor purely syntax-related ones were able to capture the quality of the machine translation output accurately, but the decomposition of WER over POS classes was the most informative one.

A sample of each language was taken, so as to aid in the comparison between manual and automatic error categorization of five error categories, namely reordering errors, inflectional errors, missing and extra words, and incorrect lexical choices. Both Spearman’s $\rho$ and Pearson’s $r$ showed that there is a good correlation with human judgment with values above 0.9.

Finally, based on the results of this error categorization, automatic post-editing rules were implemented and applied, and their performance was checked against the sample, and the rest of the data set, showing varying results. The impact on the sample was greater, showing improvement in all metrics, while the impact on the rest of the data set was negative. An investigation of that, alongside the fact that correction was not possible for Greek due to extremely free reference translations and lack of error patterns in spoken speech, reinforced the belief that automatic post-editing is tightly connected to consistency in the reference translation, while also proving that in machine translation output handling, potentially more than one reference translations would be needed to ensure better results.
# Contents

Acknowledgments ............................................. 5

1. Introduction ............................................. 6
   1.1. Purpose ........................................... 6
   1.2. Outline ......................................... 7

2. Background ............................................... 8
   2.1. Machine Translation ................................ 8
       2.1.1. Rule-Based Machine Translation .............. 8
       2.1.2. Statistical Machine Translation .............. 8
       2.1.3. Neural Networks and Neural Machine Translation ....... 9
       2.1.4. Google Translate .............................. 10
       2.1.5. Comparison of NMT to Other MT Methods ....... 10
   2.2. NLP-Cube .......................................... 11
   2.3. Evaluation of MT Output ............................ 11
       2.3.1. Human Evaluation ............................. 11
       2.3.2. Error Categorization .......................... 12
       2.3.3. Automatic Evaluation ......................... 13
       2.3.3.1. Common Metrics ............................ 13
       2.3.3.2. Syntax-Oriented Evaluation Metrics ......... 16
   2.4. Automatic Post-Editing ............................. 18
   2.5. Convertus ......................................... 19

3. Data .................................................... 20
   3.1. Swedish to English (StE) .......................... 20
   3.2. Greek to English (GtE) ............................ 20

4. Method .................................................. 21

5. Evaluation ............................................... 23
   5.1. Common Metrics .................................... 23
       5.1.1. BLEU ......................................... 23
       5.1.2. METEOR ....................................... 23
       5.1.3. TER .......................................... 23
       5.1.4. Results and Short Discussion .................. 23
   5.2. Syntax-Oriented Metrics ............................ 25
       5.2.1. POSBLEU ...................................... 25
       5.2.2. Decomposition of WER on POS classes ........... 26
       5.2.3. WPF .......................................... 27
       5.2.4. Results and Short Discussion .................. 28
   5.3. Error Categorization ................................ 31
       5.3.1. Manual Error Categorization .................. 32
Acknowledgments

I am grateful to my academic supervisor Eva Pettersson, for sharing her knowledge with me, answering all my questions, and tirelessly correcting all of my typos. Special gratitude to my supervisors at Convertus, Anna Sägvall Hein and Sebastian Schleussner, for taking me on this thesis project, and providing me with valuable data, tools, and most importantly expertise.

I would also like to thank my partner, family, and friends for being helpful, patient, and understanding with me throughout this period.
1. Introduction

Neural networks (NN) have been studied and are being used in the field of computational linguistics, thus evolving a lot. In machine translation (MT), one can come across various techniques, from rule-based (RB), to statistical (SMT), to example based, and finally to approaches using the current ever so popular neural models for neural machine translation (NMT). No matter how advanced the procedures that are used are, though, there are always ungrammatical structures in the output, such as varying verb inflections or lack of source information, such as negations. Evaluating the output of this procedure and detecting errors (also known as quality estimation, QE) is undoubtedly a tiring and time-consuming activity for human evaluators. This is why one can observe a shift from manual human evaluation to a more automatic approach that allows re-use and faster results. The same effort is being put into the process of automatic post-editing (PE), which is considered vital in MT. Manual post-editing is considered tiring since MT systems tend to repeat the same mistakes over and over again. This is why it is more desirable to develop automatic post-editing mechanisms that will be informed of the errors detected and thus, result in a good output quality, while also minimizing the human factor in the procedure.

The thesis project was carried out at Convertus AB, a machine translation services company in Uppsala, Sweden, which was created as a spin-off from Uppsala University. Convertus AB offers complete machine translation services for quality output, including automatic language checking and machine translation, as well as automatic and manual post-editing of the translated text.

1.1. Purpose

The purpose of this thesis is to evaluate and detect errors in the output of neural machine translation from Swedish and Greek, to English, using different kinds of metrics, and to formulate and apply automatic post editing rules, so as to check the impact of them on the output quality. The tool NLP-Cube1 will be used to aid in the procedure of detecting ungrammatical structures, such as varying verb inflections, or lack of source information, such as negations, etc. Common quality metrics are to be calculated for the output, and then they are to be compared to syntax-related metrics, based on the part-of-speech (POS) information derived by the tool. After these errors are categorized, they are then passed to post-editing mechanisms which will be formulated to achieve a higher quality output.

There are several research questions that this thesis wishes to address throughout the tasks:

1http://opensource.adobe.com/NLP-Cube/index.html
• How much do common evaluation metrics actually represent the quality of the MT output?

• How can syntax aid in the process of evaluating MT output quality?

• What types of errors exist in neural machine translation output? How can they be categorized manually? What about automatically? How do they differ for the Swedish-to-English and Greek-to-English pairs?

• What types of rules could be formulated for the automatic post-editing step? Can they be common or do they have to be language specific?

• What is the effect of these rules to the quality of the output?

1.2. Outline

The outline of the thesis is the following. In section 2, a background of the study is given, which discusses neural machine translation and its comparison to statistical machine translation, followed by an introduction to NLP-Cube, the tool used in this study for grammatical analysis. After that, studies proposing methods for error analysis based on POS-tags will be proposed, and common metrics for evaluation that are heavily used will be discussed. Following that, automatic PE will be presented. In section 3, the data used will be presented alongside the methods used, with the results being shown in section 4. An in-depth discussion of the results will be presented in section 5 followed by automatic PE efforts and their results in section 6. Finally in section 7, some suggestions for improvement and some guidance for future work will be given.
2. Background

2.1. Machine Translation

Machine translation (MT) is the process of translating text from one language to another using a computer, and it has always been a growing area in the field of computational linguistics, dating back to the 1950’s. In its base form it substitutes words from the source language to the target language. As one can easily understand this cannot guarantee good translations, since a language does not only consist of words put one after the other. For this reason a number of approaches to MT have been studied, and are explained in the section to follow.

2.1.1. Rule-Based Machine Translation

Rule-based machine translation (RBMT) is a method based on grammar and linguistic rules that also requires a bilingual or multilingual lexicon for the translation to be carried out (Sreelekha et al., 2018). RBMT first analyses the source sentence, then transfers the syntactic structure of the sentence using morphological and syntactic rules, and finally generates the target sentence, trying to preserve the meaning. If, for example, a Greek sentence was to be translated into an English one, a dictionary from Greek to English would be needed, as well as grammatical and syntactic rules of the two languages.

This approach requires a huge amount of manual work and domain knowledge for the rules to be gathered, prepared, and implemented. This also means that it will be time consuming and expensive to develop, as well as limited in its capacity to translate.

2.1.2. Statistical Machine Translation

Statistical machine translation (SMT) works by calculating the conditional probability that a specific sentence or word of the target language is the sentence or word of the source language. This means that words with a higher probability will result in a better translation (Sreelekha et al., 2018). The procedure followed in an SMT model is:

- parallel corpus preparation (sentence and word alignment, phrase extraction)
- bilingual translation models and monolingual language models training (a model learns and builds statistical tables)
- decoding (the target language sentences are decoded using the extracted phrases, trained translation model and language model)
- testing (of the model on unseen data)
Even though it is often highly accurate and easy to train an SMT model, parallel corpora do not exist for all languages, especially low-resource ones, and creating such corpora from scratch is time and money expensive. SMT models usually also fail to translate correctly casual style, slang words or idioms if they are not defined in the training data.

2.1.3. Neural Networks and Neural Machine Translation

A neural network (NN), as its name suggests, is a network of (artificial) neurons, which was inspired by the way a biological neural network works to process information. They have been integrated in many applications such as speech recognition, information retrieval, image analysis, classification, or data processing.

Neural networks are typically composed and organized by layers, which are made up by a number of interconnected nodes with an activation function. The input layer is responsible for presenting the patterns to the network, which are then given to the number of hidden layers to process through weighted connections. The weights can be modified by a learning rate all of them incorporate, according to the input and the patterns detected. Finally the output layer presents the process final product. An example of a NN is given in Figure 2.1.

![Figure 2.1: A NN with two hidden layers](image)

The arrows that connect the layers represent the neurons, so as to reach the prediction of the model in the output layer.

There are many types of NN, with some known examples being:

- **recurrent neural networks** (RNN) whose connections between the nodes form a directed graph, and they are suitable to use in speech recognition

- **long short-term memory** (LSTM) units which are composed of input, output and forget states to deal with the information in the cell and

- **deep convolutional networks** (DCN) which use perceptrons and require little pre-processing, heavily used in image recognition

Neural approaches to natural language processing (NLP) tasks were first introduced during the 1990’s, with NMT starting to get attention around 2004 and
being adopted by various deployments. For many they are considered the new norm since they outperform previous MT model benchmarks and their shortcomings, requiring less memory for training and promising better results. What it (very) generally does is to take inputs and predict outputs. More specifically, first the input is analyzed (encoding) and transformed into vectors and then these vectors are decoded into the target output. This is the simple encoder-decoder method that can be enriched with many approaches such as adding linguistic information, more languages, or an attention mechanism for long sentences, all to ensure better performance (Koehn, 2010; 2009;). One of the shortcomings of NMT is the big amount of data needed in the training for good performance to be expected, which is not always easy to find.

2.1.4. Google Translate

As an example one could think of the most known MT model used these days, that of Google’s translating service, which was switched from a phrase-based machine translation model to a NMT one. As Wu et al. (2016) show in their paper, the quality of the output outperforms all other baselines. Apart from the common evaluation metrics (e.g. BLEU) they use a human gold standard which points to a performance of an average bilingual human translator.

The advantage of NMT is that it is able to learn directly from the mapping of the source input to the target output. Its main weaknesses though are its slow training, slow inference speed, difficulty when rare words are encountered, and inability to translate all words of the source input (Wu et al., 2016). This is why many studies have compared NMT to previous models to see whether it is indeed a state-of-the-art method to be used.

2.1.5. Comparison of NMT to Other MT Methods

Castilho et al. (2017) investigate the use of NMT for different domains and different language sets, using automatic evaluation and human evaluation. The results varied with automatic evaluation favoring NMT, while human evaluation was more favorable towards the output of statistical machine translation. Bentivogli et al. (2016) on the other hand compare the quality of NMT output to SMT, and more specifically to phrase-based machine translation (PBMT). Their result generally showed that NMT outperforms PBMT in the post-edit effort required, for all sentence lengths, on lexically rich texts, and produces less morphology errors, lexical errors, and less word order errors. Finally, Burchardt et al. (2017) performed a linguistic evaluation on three MT engines, arguing that testing the quality of an engine based on error rates leads to more useful insights that can help improve the system. Their most important results, apart from the fact that PBMT and NMT performed the best in different categories, were that Google’s previous PBMT system was outperformed by the new NMT, while also noting that for some categories the output of NMT was similar to that of PBMT.
2.2. NLP-Cube

NLP-Cube\(^1\) is an open-source, state-of-the-art NLP tool written in Python, that can perform sentence splitting, tokenization, compound word expansion, lemmatization, tagging and parsing using Neural Networks with support for languages which are included in the UD Treebanks\(^2\). POS-tagging is done using a two-layer bidirectional LSTM, with a regularization step after the first layer. The results obtained for POS-tagging were all above the state-of-the-art ones for the task. It is stated by the researchers that NLP-Cube will help machine translation, and there is also the hope for machine translation tasks to be integrated in the tool (Boros et al., 2018).

An example output of the tool is Figure 2.1.

![Figure 2.1.: Example NLP-Cube output](image)

As we can see the format is the CoNLL-U\(^3\) format. The following information is given:

- first column: word index for each new sentence,
- second column: word form or punctuation symbol,
- third column: lemmatized or stemmed word,
- fourth column: universal POS-tag (UPOS),
- fifth column: language specific POS tag,
- sixth column: morphological features (FEATS),
- seventh column: the head of the word in the sentence,
- eighth column: universal dependency relation to the head of the sentence.

2.3. Evaluation of MT Output

2.3.1. Human Evaluation

Generally, and as Sun (2010) also claims, in the task of translating there can never be one truly ‘correct’ translation due to differences in the translator style, vocabulary range, and domain specialization. Despite that, for evaluating (N)MT output there are a number of ways to do so, by using various means. One way to

---

2. [https://universaldependencies.org](https://universaldependencies.org)
3. [https://universaldependencies.org/format.html](https://universaldependencies.org/format.html)
achieve evaluation is through human (native speaker) evaluation, which is mainly based on adequacy and fluency, with the first dealing with how much content and meaning from the original text is also present in the output, while the latter is concerned with whether the output is fluent regarding the target language, mainly dealing with grammatical correctness and idiomatic expressions. That is why Sun (2010) examines how four translators can rank various translations of a document from 1 to 4, using these results to then correlate these rankings to three automatic metrics of MT evaluation, namely BLEU, TER and GMT (General Text Matcher which is an implementation of precision and recall).

Bremin et al. (2010) in their paper compare four evaluation methods on output derived from three different MT systems. The main comparison is done between human error analysis and automatic metrics which are common ways to perform evaluation, and reading comprehension and eye-tracking, which are less discussed. It was shown that for large quality difference between the systems, the latter two approaches give similar results to the more common ones.

The use of eye-tracking as an evaluation metric was also investigated by Guzmán et al. (2015). Their comparison criteria was the background of the people evaluating as well as the information given to them during evaluation in different parts of the screen, which included the hypothesis sentence, the source sentence, the context of the source sentence, the reference translation, and the context of the reference translation. Their results showed that between monolingual and bilingual people, monolingual people took longer to evaluate translations, except for when information of the target sentence was provided, but at the same time were more consistent in their evaluation. It was also proven that the more information given to the translators, the more time the task needs in order to be completed.

2.3.2. Error Categorization

Apart from fluency and adequacy, what can be considered more important in an evaluation is not how many mistakes there are, but what types of mistakes. For this a (weighted) categorization of the errors could be made to be able to determine how much they affect the quality of the output.

Popović (2017) in her paper presents an error categorization for German to English errors in SMT and NMT. For NMT output it is claimed that it is better at handling issues related to word order, morphology and fluency, such as verb order, articles and phrase structure, etc. It is also noted that the method’s main weaknesses are prepositions, ambiguous word translation from English to German, and the continuous tense of English.

In another paper by Stymne (2013), an error typology based on a Swedish grammar checker is defined, so as to be used for error categorization of MT output. It included fluency and adequacy errors and it showed how all errors were classified into categories, and how verb form errors were in need of additional categories so as to be classified correctly. The comparison was made against adult and children evaluators, with the method of the paper achieving the highest precision, lower than the adults’ recall, but the same as the children’s. It is believed that this method will be useful for annotating MT text, and also an idea of automating the procedure is introduced.
BLAST (Stymne, 2011) is an example of a tool for automatic error analysis of MT output. It is written in Java and can help MT evaluation projects through an easy-to-use graphical user interface through which one can add, edit, or search existing annotations of the text or even have highlighted the similarities of the output to a reference translation. Its purpose is to be flexible so as to not be tied to specific systems, languages or error typologies.

2.3.3. Automatic Evaluation

2.3.3.1. Common Metrics

Even though human evaluation methods show good results at times, it is generally believed that an automatic evaluation method that has a high correlation to human judgment is preferable since it is considered less time consuming, less expensive, and not language bound. Using these kinds of methods ensures less human labour, and the ability to re-use the method with minimal cost. The most known and commonly used metrics for MT evaluation are BLEU (Papineni et al., 2002), METEOR (Denkowski and Lavie, 2011) and TER (Snover et al., 2006) discussed in this section.

Papineni et al. (2002) were the first to formulate such a metric, called BLEU, (ranges from 0 to 1) based on the idea that for such a metric to be useful, it would need to be close to a human translation(s) based on a numerical metric, namely the word error rate (WER) metric. It can also be used with many translations used as reference, which is considered optimal, since, as we have mentioned again, more than one translation can be considered correct for one source sentence.

The metric is based on the unigram precision metric, which for a hypothesis translation \( X \) and a reference translation \( Y \) means:

\[
\text{Precision} = \frac{X \cap Y}{X}
\]

As it is easy to guess, this means that when a hypothesis word appears more than one time in different references, it will be counted more than one time as well, which can result in high precision that does not denote a good translation. As an example, one can think of the following (Papineni et al., 2002):

- **ref 1**: The cat is on the mat
- **ref 2**: There is a cat on the mat
- **hyp**: The the the the the the the

Basic unigram precision for this hypothesis sentence would be \( 7/7 = 1 \), since all the words of the hypothesis appear in the reference sentences.

This is why the researchers propose the calculation of the modified n-gram precision, which is essentially to count the maximum total count of each word in the candidate in all reference translations. Then the count of each word is clipped to a maximum of that word, which then is summed over all the words of the candidate, and then divided by the total number of n-grams in the hypothesis. This can be used for all n-grams, and not only unigrams, by making use of the geometric mean. Taking the before mentioned example again, the modified
unigram precision now would be, 2/7, which is a more suitable way of scoring the translation.

Another translation problem tackled by Papineni et al. (2002) is the precision penalty assigned to candidate sentences that are longer than their reference sentences. For this reason they introduced the **brevity penalty**, which takes word choice and not word count into consideration so as to not allow short candidates to have a high score. This is done for the whole corpus by first calculating the total number of best matches, followed by the sentence length of each hypothesis sentence, and then perform a division over the length of the whole hypothesis translation:

\[
BP = \begin{cases} 
1 & \text{if } c > r \\
\varepsilon^{(1-r/c)} & \text{if } c \leq r.
\end{cases}
\]

where \(BP\) is the brevity penalty, \(c\) is the length of the hypothesis sentence, and \(r\) the length of the reference sentence(s).

Then for the BLEU score to be derived the geometric average is employed, with n-grams equal to 4 and uniform weights summing to one:

\[
\text{BLEU} = BP \cdot \exp \left( \frac{1}{N} \sum_{n=1}^{N} w_n \log p_n \right).
\]

where \(BP\) is again the brevity penalty, \(w_n\) are the weights, and \(p_n\) is the geometric average of the unigram precision.

One drawback of the metric is that for sentence level it has a low correlation to human judgment due to the calculation of the geometric mean precision, so if a "higher order n-gram precision of a sentence is 0, then the BLEU score of the entire sentence is 0, no matter how many 1-grams or 2-grams are matched" (Chen and Cherry, 2014). For this reason 7 smoothing techniques were investigated, to enable BLEU to perform better on sentence level. All of the proposed smoothing techniques had a positive effect in BLEU calculation on sentence-level, with three novel ones proposed by Chen and Cherry (2014) resulting in higher correlation than the already existing 4 ones.

**METEOR** (Denkowski and Lavie, 2011) was then created to address some of the shortcomings of the before-mentioned metric, and it was shown to outperform it regarding correlation to human judgment. It is based on the harmonic mean between precision and recall, giving a higher weight to precision. It computes a score based on "explicit word-to-word matches between the translation and a given reference translation" (Denkowski and Lavie, 2011), and if there exist more than one reference, each score is calculated, and the best one is used. It also supports stemmed word matching and synonymous word matching, as well as parameter tuning so as to enable better results for different languages. The formulas for calculating weighted precision, recall and the parameterized harmonic mean respectively are the following:
Content and function words are represented by $h_c$ and $h_f$ respectively for the hypothesis and $r_c$, $r_f$ for the reference. $m_i$ is each of the matchers, so $m_i \cdot c_h$, $m_i \cdot h_f$, and $m_i \cdot r_c$, and $m_i \cdot r_f$ are the number of these words that appear in the matches. $w_i...w_n$ are the weights of the matches and $\delta$ is the weight of the word.

The fragmentation penalty assigned by METEOR is calculated as follows:

$$
Pen = \gamma \cdot \left( \frac{ch}{m} \right)^{\beta}
$$

In the above formula $ch$ is the number chunks.

This penalty takes into account differences in word order, and finally the METEOR score is calculated as:

$$
Score = (1 - Pen) \cdot F_{\text{mean}}
$$

$\alpha$, $\beta$, and $\gamma$ are parameters that can be tuned to achieve higher correlation to human judgment.

As in BLEU a higher METEOR score indicates a better translation quality for the MT output.

Finally translation error rate, TER, (Snover et al., 2006) was created as a way to measure the quality of an output based on meaning, something that is closer to human judgment. It is written in Java and it achieves this by calculating the amount of post-editing required in regards to a reference translation as in the following:

$$
\text{TER} = \frac{\text{number of edits}}{\text{average number of reference words}}
$$

The edits mentioned in the formula are substitution, deletion, insertion and word shifts, with the same cost (1) for each. The first three edits are calculated using dynamic programming, while the last one is calculated using greedy search. This is done so that first the shifts are calculated so as to minimize the number of insertion, deletions and substitutions necessary until no more shifts are left to be done.

An example of TER calculation is the following (Snover et al., 2006):

**REF:** SAUDI ARABIA denied THIS WEEK information published in the AMERICAN new york times

**HYP:** THIS WEEK THE SAUDIS denied information published in the new york times
Applying TER to the hypothesis and reference given above, we identify 4 edits, namely 1 shift ("this week"), 2 substitutions ("Saudi Arabia"->"the Saudis"), and 1 insertion ("American"), which gives a score of 31%.

Since this metric is based on errors it is different than the other two, meaning that a small TER value indicates better quality.

The problem of any of these kinds of metrics still remains. A sentence can be translated correctly in many different ways, and thus comparing MT output to some human translated outputs will lead to shortcomings.

2.3.3.2. Syntax-Oriented Evaluation Metrics

These kinds of automatic metrics, as the ones described in 2.3.3.1, do not necessarily point to the quality of the MT system, in the sense that they do not carry information regarding the nature and types of errors, and a correspondence between them and the errors is difficult to find. This is why many researchers have put an effort into evaluating MT output with the use of different linguistic features, either extending already existing metrics to include these features, or following a different route altogether.

Tezcan et al. (2016a) in their paper describe a method of detecting grammatical errors in Dutch-to-English MT output using dependency parsing and tree-bank querying. A dependency tree is a directed acyclic graph, which represents all the relations of the words in a sentence. So, as a first approach when no parse covers all the input, it is marked as an indicator of an error. As a second approach the subtrees of a sentence are queried against a gold standard by "using dependency relation and syntactic category information on phrase and lexical level" (Tezcan et al., 2016a), with matching constructions indicating errors. Both approaches were tested on a word and sentence level, and it was proven that they perform highly accurately on the sentence level, especially when combined.

In Popovic et al. (2006) a new framework is proposed for automatic error analysis based on morpho-syntactic information, between Spanish and English. They tried to tackle noun and adjective related syntactic differences, as well as verb, adjective and noun inflection errors between the languages. Their results showed that this automatic method performed as well as human evaluation.

Following this paper, Popovic and Ney (2006) worked specifically on error analysis for verb inflections for Spanish. For example, it can be the case that the same form of an English verb can correspond to two different Spanish forms. The metrics used were position independent error rate (PER) and word error rate (WER), and mainly precision, recall and f-measure based on PER, for each verb inflection. It was noted though that since there is no one-to-one correspondence in the translation task, these metrics are not an exact equivalent to precision, recall and f-measure. PER-based recall (perR) basically indicates how many verb inflections in the reference are found, and is calculated as follows:

- all verb forms of the verb type are extracted from the reference
- each verb, whose base form occurs in the hypothesis sentence that corresponds to it, is extracted from the hypothesis
- PER is calculated and subtracted from 1, giving the perR
PER-based precision (perP) indicates how many verb inflections in the hypothesis are translated correctly and is calculated as follows:

- all verb forms of the verb type are extracted from the reference
- each verb, whose base form occurs in the hypothesis sentence that corresponds to it, is extracted from the reference
- PER is calculated and subtracted from 1, giving the perP

and finally PER-based F-measure (perF) is the standard harmonic mean between the two before mentioned ones, indicating how difficult the translation of the inflection is, and is calculated as follows:

\[ perF = \frac{2 \cdot perR \cdot perP}{perR + perP} \]

In the results, PER-based f-measure shows what types of inflections are hard to translate, while PER-based precision and recall graphs indicate which inflections tend to be wrongly translated.

Popović and Ney (2007) in another paper, deal with the use of POS-tags and decomposition of WER and PER, and additionally how these could be used in automatic error analysis to estimate the number of inflectional errors and also find the distribution of missing words regarding POS classes. It was shown that the results correspond to results obtained by human analysis.

Finally, Popović and Ney (2009) propose some evaluation metrics that are syntax oriented, extending common ones like BLEU, METEOR, and TER to use syntactic information so as to strengthen them. The metrics proposed are calculated on POS n-grams and not words, and are

1. **POSBLEU** (BLEU score calculated on POS tags)
2. **POSP** (number of POS n-grams in the hypothesis with a corresponding POS-tag in the reference)
3. **POSR** (number of POS-tags in the reference that are also present in the hypothesis)
4. **POSF** (all POS-tags that have a corresponding one both in reference and hypothesis) and
5. **WPF** (an F-measure that takes into account both words and POS-tags)

It was tested and proven that these metrics correlate well with human judgment, regarding both adequacy and fluency, with POSBLEU and POSF scores being the better performing ones, and with WPF not performing as well, but nevertheless having the advantage of using both words and POS-tags.
2.4. Automatic Post-Editing

Post-editing (PE) is a necessary step of MT to enable the output to have the highest possible quality. This step is usually done manually by human ‘post-editors’, but it is a labour intensive task, so automatic ways to carry it out are more desirable. Many approaches have been, and are being, used for automatic PE, including regular expressions, statistical methods, or even neural methods.

Kjellin (2012) in his thesis presents a method for generating post-editing candidate rules from a parallel MT output corpus and manually post-edited corpus. Then this set is filtered through applying these rules to another corpus and using BLEU, TER and METEOR, the ones that are able to increase the scores are kept. The results showed that the method proposed in the paper is able to increase the scores for translations derived from the same domain as the corpus used for filtering. The impact of PE rules was also investigated by Mostofian (2017) in her thesis. After using BLEU and TER to estimate the quality of the translation and analyzing the results, categories of errors were identified so as to aid in writing the automatic PE rules. For the largest data set of the study the results of the experiments showed an improvement of the BLEU score.

Quality estimation (QE) and PE were also combined by Chatterjee et al. (2018). There are three different strategies used in the paper, regarding the role of QE and PE. QE can either be:

1. an activator (when the quality is below a certain threshold, the PE is activated)
2. a guide (QE labels are used to guide PE rules on which tokens need change) or
3. a selector (QE predictions are used to select the best solution between the raw MT and its automatically corrected version)

These strategies were tested, and the results showed that the second and third strategy resulted in an improvement in PBMT output.

Rafael Guzmán in his work (Guzmán, 2008); (Guzmán, 2007) also investigated automatic ways of PE using regular expressions and linguistic patterns between English and Spanish. These linguistic patterns were used to identify errors that occurred systematically in the MT output, then using regular expressions he proposes to match these errors, and subsequently replace them with the correct token. The main errors he identified were misspellings, punctuation, articles, prepositions, grammatical agreement, word order, reflexive pronouns, style, and redundancies. It is stressed that these patterns should be first grouped by not only frequency but also relevance, so as to ensure a good performance.

Guzmán (2008), further analyses the use of regular expressions, but this time this use is mainly focused on the task of disambiguation in translation. This is done by enriching the regular expression with context information. He then proceeds to make very precise tuning to the regular expressions so as to deal with mistranslation of -ing, mistranslation of subordinate clauses, and mistranslation of verbs with ambiguous meanings from English to Spanish. The results point to the correlation of predictable patterns in MT output to the good performance of regular expressions in the task of PE.
2.5. Convertus

Convertus AB offers translation systems through an in-house web application, which customers can use by uploading their documents (a variety of forms is supported), to be translated by one of the engines (RBMT, SMT, NMT). The user can then edit/polish the translation through an editing interface. Since the application keeps track of all the steps in the translation, the user is able to keep track of multiple documents. Included is also a translation memory, which saves previously seen sentences. Some of the services include Convertus BTS, and SDL Trados Studio.
3. Data

3.1. Swedish to English (StE)

One of the corpora used in this study contains data collected from 1177.se, which is a webpage providing information on health-care services for individuals residing in Sweden. It is also available in 10 other languages, including English. The corpus was created from the website around 2016, and it consists of four txt files of the most common Swedish sentences, and one csv file which included 2217 Swedish-to-English sentence pairs, and is the one used in this paper. The data are publicly available and the before mentioned files were provided by Convertus AB, and where split into two different documents, the source sentences and the reference sentences. The source sentences were passed through Google’s API so as to acquire the hypothesis sentences. Google API services were provided by Convertus AB.

3.2. Greek to English (GtE)

The other corpus used is the Europarl parallel corpus, a corpus of the European Parliament proceedings for more than 21 languages. As described in Koehn (2004), for the parallel corpora to be created first matching items were found and labeled with the corresponding document IDs. Then sentence boundaries were identified in pre-processing and finally the sentences were aligned using a tool based on the Church and Gale algorithm (Gale and Church, 1993). The corpus is free for download and use. After the documents were acquired, a specific parliament session was chosen, from which 2217 random sentences were extracted, to follow the size of the Swedish data.
4. Method

The purpose of this study is not only to compare commonly used metrics in MT output evaluation to syntax-aided ones, but also categorize the errors in the output and correct them through APE rules.

The following flowchart depicts the procedure that will be described in the following chapters, which is explained shortly after the figure.

After the data are all gathered, the source language sentences will be run through Google’s API to be translated. Then each language pair will consist of the following:

- a txt file containing the source language sentences,
- a txt file containing the hypothesis sentences (acquired through Google’s NMT), and
- a txt file containing the reference sentences (to be used for evaluation)

Homogeneity is ensured since the outputs will be derived by the same MT method, have the same size.
Following that BLEU, METEOR and TER scores will be calculated, so as to judge the quality of the translation. As a comparison, syntax-related metrics, namely POSBLEU, WPF and WER over POS classes, will also be calculated to gather more information on the types of errors. To do that first, NLP-Cube is used on the data, to enable the use of various syntactic information, as visible in Chapter 2.2.

The data will then be divided for the error categorization task, where for each language pair, 200 sentences will be randomly taken so as to allow for the creation of a development set which will be representative of the whole set. From these 200 sentences errors will be targeted and categorized so as to be used in the automatic post-editing step. After the APE rules are implemented, their impact will be noted by re-evaluating the quality of the translation on the sample data set and the rest of the data set.
5. Evaluation

5.1. Common Metrics

5.1.1. BLEU

After acquiring all data necessary, we begin by automatically calculating the quality of the translation we obtained using the BLEU score. A score of 1 indicates a perfect translation, while a score of 0 indicates a bad translation. As explained in 2.3.3.1, the BLEU score is more accurate when a smoothing technique is applied to it. For this experiment the fifth smoothing technique as explained in Chen and Cherry (2014) was used, which was a novelty to their paper. It is based on the idea that "matched counts for similar values of n should be similar" (Chen and Cherry, 2014), and this is why it works by averaging the n-1, n and n+1 grams to calculate the matched count.

For both the BLEU score calculation and the smoothing technique NLTK’s (Bird and Klein., 2009) BLEU score from the translate module was used alongside the SmoothingFunction() module¹.

5.1.2. METEOR

Calculation of METEOR was done using a tool written in Java². As was mentioned in 2.3.3.1, this metric is more informative in calculating translation quality than BLEU. As with BLEU though, the lower the score, the worse the quality of the translation.

5.1.3. TER

Finally, to calculate TER the general algorithm used is depicted in figure 5.2. For this experiment with TER, the tercom³ program was used (Snover et al., 2006). As was mentioned in 2.3.3.1, and as opposed to BLEU, the lower the TER score, the better the quality of the translation, since it means that less edits were calculated.

The results obtained are discussed in the following section.

5.1.4. Results and Short Discussion

Table 5.1 shows the automatic evaluation scores of BLEU, METEOR, and TER obtained after the source sentences were translated using Google’s API services.

¹https://www.nltk.org/_modules/nltk/translate/bleu_score.html
²https://github.com/cmu-mtlab/meteor
³https://github.com/jhclark/tercom
As one can see from these preliminary results, all three metrics denote an unsatisfactory translation output in relation to the reference given, especially given the fact that it is believed that they correlate well with human judgment. For the GtE language pair we can see that the performance is slightly lower than the StE one, by some points. It is also visible that even though the scores do not denote a good translation quality in general, the more information the metric includes in its calculation of the translation, the more one can infer about the translation output.

For the GtE language pair, for example, we can see a slight difference in the BLEU and METEOR score, thus, one could argue that METEOR does a better job at depicting the quality of the translation since it takes into account features like stemming or synonyms. This is also the case for the StE pair, where we can see that the BLEU score is small (thus denotes a bad translation quality), and METEOR has a higher score than BLEU, which, nevertheless, also denotes errors in the translation hypothesis. If once combines this with the relatively high TER scores, then one can infer that the quality of the MT output is poor.

It should be noted though, that these metrics work better when multiple references are given. So the low scores and the difference between them could also be attributed to the fact that only one reference was provided for each sentence.
5.2. Syntax-Oriented Metrics

The before-calculated metrics, even though they denote the general quality of the translation, are not able to provide information on the kinds of errors present in the output, only the number. For this reason, in this section, new metrics will be calculated, based on syntax that will be able to provide more clues on the kinds of error that exist in the output, to enable us to target, group them, and correct them later on.

This was achieved through the use of the tool NLP-Cube, as presented in 2.2, on both the hypothesis and the reference data, which enabled us to gather all syntactic information from the CoNLL-U format which is output by the tool.

It is of course expected that the use of syntactic information would potentially introduce more noise into the calculation of the metrics since most tools cannot perform well on erroneous MT output, but since this tool is state-of-the-art with good performance compared to other baselines, it is believed that these errors will not have a great impact on the calculation. For reference, table 5.2. shows the performance of the tool on four different Universal Dependencies English corpora (en_ewt4, en_gum5, en_lines6, and en_pud7), for all the tasks it can perform (Boroș et al., 2018).

<table>
<thead>
<tr>
<th>Language</th>
<th>Tok</th>
<th>SS</th>
<th>Word</th>
<th>Lemma</th>
<th>UPOS</th>
<th>XPOS</th>
<th>Morpho</th>
<th>CLAS</th>
<th>BLEX</th>
<th>MLAS</th>
<th>UAS</th>
<th>LAS</th>
</tr>
</thead>
<tbody>
<tr>
<td>en_ewt</td>
<td>99.26</td>
<td>76.32</td>
<td>99.26</td>
<td>94.51</td>
<td>95.25</td>
<td>94.83</td>
<td>96.03</td>
<td>79.31</td>
<td>73.77</td>
<td>73.75</td>
<td>85.49</td>
<td>82.79</td>
</tr>
<tr>
<td>en_gum</td>
<td>99.65</td>
<td>82.13</td>
<td>99.65</td>
<td>91.70</td>
<td>94.71</td>
<td>94.42</td>
<td>95.64</td>
<td>75.01</td>
<td>65.38</td>
<td>67.42</td>
<td>84.10</td>
<td>80.59</td>
</tr>
<tr>
<td>en_lines</td>
<td>99.91</td>
<td>87.80</td>
<td>99.91</td>
<td>93.89</td>
<td>96.38</td>
<td>95.01</td>
<td>96.46</td>
<td>74.70</td>
<td>67.82</td>
<td>69.28</td>
<td>82.58</td>
<td>78.03</td>
</tr>
<tr>
<td>en_pud</td>
<td>99.74</td>
<td>95.70</td>
<td>99.74</td>
<td>94.32</td>
<td>95.14</td>
<td>93.88</td>
<td>94.99</td>
<td>82.36</td>
<td>76.12</td>
<td>72.76</td>
<td>88.27</td>
<td>85.31</td>
</tr>
</tbody>
</table>

Table 5.2.: NLP-Cube performance on English data

5.2.1. POSBLEU

The first syntax-oriented metric calculated is the POSBLEU. As its name suggests its main idea is the calculation of the classic BLEU score on the POS tags of the words instead of the words themselves. As Popović and Ney (2009) show in their article, this novel metric was one of the most promising ones introduced and it correlated well with human judgment on the document level, but not as high on the sentence level.

POS BLEU, along with other metrics, was compared to human judgments using the Spearman’s correlation coefficients $\rho$, which describes the relation between two variables using a monotonic function. This means that it will be high if the variables are similar, and low otherwise. When comparing human judgment to automatic translation metrics, generally, the same holds, meaning the higher value of the automatic metric, the more correlated it is to the human judgment. Experiments by Popović and Ney (2009) showed that POSBLEU scored 0.642 and 0.626 (scale: 1 to -1) on adequacy and fluency respectively, and 0.712 (scale: 1 to -1) on the sentence ranking, where human evaluators were asked to rank

---

5 https://universaldependencies.org/treebanks/en_gum/index.html
7 https://universaldependencies.org/treebanks/en_pud/index.html
translated sentences which were relevant to one another. It was the highest value acquired in the paper among all the introduced metrics, and it is why this metric is used here.

To calculate POSBLEU, the same tool and the same smoothing technique was used as in 5.1.1. The input were two files where each sentence of the hypothesis and the reference documents was replaced with the POS tags (UPOS tags plus morphological features) of each word, derived from running the documents through NLP-Cube.

As was mentioned, the correlation of this metric to the human judgment at the sentence level was not as high as at the document level. This is explained due to the fact that mere POS tags lack lexical information.

5.2.2. Decomposition of WER on POS classes

This metric as in Popović and Ney (2007), was introduced as a novel metric, which would help identify types of errors in the MT output through the decomposition of the WER metric over different POS classes.

Word error rate (WER) in general, is a metric which counts the number of Levenshtein distance operations necessary so that the hypothesis sentence is transformed into the reference sentence as:

$$\text{WER} = \frac{S + D + I}{N}$$

where $S$, $D$, and $I$ denote the substitutions, insertions and deletions, and $N$ the number of correct words (words in reference).

For reference, it should be mentioned here that the WER score for the two data sets was:

- WER for StE: 70.58% (24090/32128)
- WER for GtE: 74.78% (56860/76028)

This new metric allows for identification of how much each erroneous word contributes to the WER score. For each POS class this metric can be calculated as (Popović and Ney, 2007):

$$\text{WER}(p) = \frac{1}{N_{\text{ref}}} \sum_{k=1}^{K} n(p, \text{err}_k)$$

where $\text{err}_k$ denotes the set of erroneous words in sentence $k$ with respect to the best reference and $p$ is the POS class. Then this means that $n(p, \text{err}_k)$ is the number of errors in $\text{err}_k$ produced by words with POS class.

As an example one can think of the two following sentences (Popović and Ney, 2007):

- hypothesis: “Mrs Commissioner, twenty-four hours is sometimes too much time.”
- reference: “Mister Commissioner, twenty-four hours sometimes can be too much.”
Standard WER for this sentence is 33.3% (4/12).
Now if we take the same two sentences and add their POS tags we get the following:

- **hypothesis:** "Mrs(N) Commissioner(N),(PUN) twenty-four(NUM) hours(N) is(V) sometimes(ADV) too(ADV) much(PRON) time(N).(PUN)"
- **reference:** "Mister(N) Commissioner(N),(PUN) twenty-four(NUM) hours(N) sometimes(ADV) can(V) be(V) too(ADV) much(PRON) time(N).(PUN)"

If we decompose WER over the POS classes of the sentences we get:

- **WER(N) = 1/12 = 8.3%**
- **WER(V) = 2/12 = 16.7%**
- **WER(ADV) = 1/12 = 8.3%**

As it is visible this decomposition allows for more information on the kinds of errors that exist in the sentence, as opposed to a plain score of how many errors there are (e.g. deletion edits could denote missing words).

To calculate this metric three functions were combined in a python script. First, a standard WER function is calculated, which outputs the matrix of the words instead of the WER score. After that another function uses this matrix to append to a list the words that were involved at each edit. Finally the third script takes each word of the list and replaces it with its UPOS through NLP-Cube. It should be noted that for substitution and deletion errors the POS tags of the reference were taken into account, while for insertion errors the POS tag of the hypothesis was taken into account. The POS tags are then distributed to different lists according to that POS. Finally, the length of each list is divided by the length of the reference document to obtain the desired percentage.

5.2.3. WPF

WPF was another metric introduced by Popović and Ney (2009), to counteract the before mentioned disadvantage of a syntax-oriented metric lacking lexical information. For this reason, an F-measure on both words and POS-tags is proposed. It is calculated by taking into account all words and all POS tags which have a counterpart in both the reference and the hypothesis.

According to the experiments of the paper, the correlation of this score to human judgment is not as high, as the other metrics which were purely POS-based. It does however have the advantage over them of including the lexical aspect into the calculation of the score.

To calculate WPF, and since no strict formula was provided in the paper, first precision, recall and fscore were calculated for the POS-tags and the words separately as:

\[
\text{Precision} = \frac{\text{hypothesis} \cap \text{reference}}{\text{total number of words}}
\]
\[
\text{Recall} = \frac{\text{hypothesis} \cap \text{reference}}{\text{number of words in reference}}
\]
Then these two f scores were added together and divided by two to provide the final WPF score. Simple arithmetic mean was used as well as python sets to ensure a $O(1)$ look-up complexity.

5.2.4. Results and Short Discussion

The following tables shows the results obtained in the previous section, with tables 5.3 and 5.4 representing POSBLEU and WPF scores, and table 5.5 representing WER scores over different POS classes.

<table>
<thead>
<tr>
<th>Language Pair/Score</th>
<th>POSBLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>StE</td>
<td>0.792</td>
</tr>
<tr>
<td>GtE</td>
<td>0.846</td>
</tr>
</tbody>
</table>

Table 5.3.: POSBLEU Scores per Language Pair

<table>
<thead>
<tr>
<th>Language pair /Score</th>
<th>WPF</th>
</tr>
</thead>
<tbody>
<tr>
<td>StE</td>
<td>2.086%</td>
</tr>
<tr>
<td>GtE</td>
<td>2.150%</td>
</tr>
</tbody>
</table>

Table 5.4.: WPF Scores per Language Pair

<table>
<thead>
<tr>
<th>UPOS/Language Pair</th>
<th>StE</th>
<th>GtE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADJ (adjective)</td>
<td>4.98%</td>
<td>4.70%</td>
</tr>
<tr>
<td>ADP (adposition)</td>
<td>5.50%</td>
<td>6.54%</td>
</tr>
<tr>
<td>ADV (adverb)</td>
<td>3.42%</td>
<td>4.13%</td>
</tr>
<tr>
<td>AUX (auxiliary verb)</td>
<td>5.37%</td>
<td>4.79%</td>
</tr>
<tr>
<td>CCONJ (coordinating conjuction)</td>
<td>1.33%</td>
<td>1.40%</td>
</tr>
<tr>
<td>DET (determiner)</td>
<td>5.02%</td>
<td>4.42%</td>
</tr>
<tr>
<td>INTJ (interjection)</td>
<td>0.20%</td>
<td>0.37%</td>
</tr>
<tr>
<td>NOUN (noun)</td>
<td>17.71%</td>
<td>14.30%</td>
</tr>
<tr>
<td>NUM (numeral)</td>
<td>0.33%</td>
<td>0.44%</td>
</tr>
<tr>
<td>PART (particle)</td>
<td>2.54%</td>
<td>2.53%</td>
</tr>
<tr>
<td>PRON (pronoun)</td>
<td>5.69%</td>
<td>5.64%</td>
</tr>
<tr>
<td>PROPN (proper noun)</td>
<td>1.08%</td>
<td>1.81%</td>
</tr>
<tr>
<td>PUNCT (punctuation)</td>
<td>7.74%</td>
<td>7.75%</td>
</tr>
<tr>
<td>SCONJ (subordinating conjuction)</td>
<td>1.19%</td>
<td>0.82%</td>
</tr>
<tr>
<td>SYM (symbol)</td>
<td>0.06%</td>
<td>0.04%</td>
</tr>
<tr>
<td>VERB (verb)</td>
<td>8.94%</td>
<td>8.13%</td>
</tr>
<tr>
<td>X (other)</td>
<td>0.31%</td>
<td>0.37%</td>
</tr>
</tbody>
</table>

Table 5.5.: WER Score of Various POS tags per Language Pair

As we can see the scores differ from the scores presented in 5.1.4. Starting with POSBLEU we can see that the scores present a much higher translation quality
than the regular BLEU score, but it is obvious that it follows the same pattern of
the StE pair scoring higher than the GtE pair. Even though a direct comparison
of BLEU and POSBLEU is not possible, since they are based on different grounds,
from the results one could say that the difference between them is that BLEU
denotes that the translations differ on the word level (no exact match), while
POSBLEU shows that they actually do not differ that much syntax-wise. This
means that even though the exact words of the hypothesis were not present in
the reference, their syntactic part was the same. Taking also into account that the
comparison included morphological features, one could easily infer that the types
of errors could be variations of the same word, or stylistic errors.

It should be noted here that pure POS-tags metrics like POSBLEU are still
not adequate enough. If for example the hypothesis sentence was "This apple is
tasty." and the reference sentence was "This book is nice." according to a purely
POS-based metric it would score 100%. This is why they are not informative
even to be used on their own, and this is one possible reason for the high scores
acquired above when using purely syntax-oriented measures.

For this reason WPF was also calculated in the previous section, as a com-
bination of word and POS-tag metrics, even though its correlation to human
judgment was not as high. As we can see from the results, the performance for
this metric is rather disappointing. Even though the StE scored again higher than
GtE, confirming the already existing pattern, a percentage that low could either
denote a bad correlation between POS tags and words, or a poor performing
metric.

The low performance of the metric can be explained seeing also how Lavie
et al. (2004) used unigram precision, recall, and f-measure and a weighted version
of these, and reached the conclusion that mainly recall is useful for MT systems
since it is a given that a good translation system should preserve as much of the
meaning of the input as possible. Our low scores could thus also be attributed to
low precision that penalized the f-measure score.

Finally, the third metric was chosen to provide more concrete information
regarding the NMT output, which would help with categorizing and correcting
these errors later on. The decomposition of WER over POS-classes took the two
before mentioned metrics a step ahead, seeing how the initial scoring (WER score)
is calculated on words, and then syntactic information of these words is used to
check which classes play an important role in the quality of the output. It should
be noted though, that word reordering cannot be identified by the WER score,
since it is not supported by the algorithm.

For both the StE and the GtE we see that even though all the other metrics
differed in their scoring, this one shows similar results. The most interesting and
at the same time high scored categories present in both are verbs and auxiliary
verbs, nouns and adjectives, pronouns but also punctuation.

A quick investigation behind the reason for the puzzling high percentage for
the punctuation category, by debugging the script, showed that python took as
difference the space between the sentences as well, which is not an error to
begin with, since the translation of a sentence might be larger or smaller than the
reference. The rest of the items in the list were commas and hyphens, indicating
differences in compound words and comma placement between the translations.
These are thus, not considered an important error category.
Regarding the other categories, one could think of various reasons behind the errors that might have led to this performance. For example, for Greek, which is as mentioned before an inflectional language, the noun and adjective errors cannot be attributed to that, since English lacks inflections, so no mistakes can be made there. This can only mean that this percentage shows a possible case of synonyms or word substitution errors. The verb and auxiliary verb errors on the hand might have a different explanation behind them, since third person singular inflection in Greek is translated differently than the other inflections, e.g. "πάει" should be translated into "goes", while "πάμε" or "πάω" etc. should be "go". A failure of the MT system to correctly predict these could have led to a difference between the reference and the hypothesis.

Finally, regarding pronouns, the difference between the two languages does not reflect the before mentioned issues, since there is a one-to-one correspondence between pronouns. One case where there is no exact correspondence is the ‘you’ pronoun which is used for both second person singular, and second person plural, while the corresponding Greek ones are "εσύ" and "εσείς" respectively. This cannot point to any connection to the error rate though, since both the pronouns and the inflection of the verb following those will be lost when translated into English, thus making "εσύ πάς" and "εσείς πάτε" to be both translated to "you go" with only the context being used to tell them apart. It is plausible to believe that the percentage can be attributed to a more free translation in the reference, for example "εγώ κι εσύ" (you and I) to have been translated into "we" in the reference instead.

For Swedish we observe a similar trend of verbs, auxiliary verbs, pronouns, nouns and adjectives dominating the table. Verb differences can be attributed to different use of tense between the two languages. For example, "I will go to the cinema" and "I am going to the cinema", both correspond to "Jag ska gå på bio". Regarding nouns and pronouns, it is believed that a similar trend as the one described for the Greek language holds. Since nouns in Swedish are inflected regarding singular or plural, genitive, and article, different choice of translation for nouns could have led to the percentage visible, as well as inability to correctly translate the inflection might have led to the performance visible. Finally, the explanation behind pronouns is the same as for Greek; difference in translation choice.

Even though the conclusions that can be reached by looking at all three scores and the score variations are not as concrete, one could infer that some shortcomings related to the translation, are mainly tied to the fact that even though the translations do not match on the word level, they do match on the syntax level. This is something that can lead us to believe that the issues of the translation are in accordance with the NMT usual output which is fluent (good POS metrics) but may from untranslated words, omission of words in long sentences, etc (bad word metrics). Also, from the three metrics presented here, it is believed that the decomposition of WER over different POS classes is the most informative one, and we plan to use it for automatic error categorization later in the thesis.

Having thus scored and acquired a very spherical idea of the quality of the translations and what kinds of errors this translation might contain, we move on to categorizing those.
5.3. Error Categorization

For this part of the thesis, 200 (roughly 10% of the data) random sentences were picked from each language pair, and from each of the three files it contained (original sentences, reference sentences, hypothesis sentences). The random sample was generated so as to contain only unique values, so that no two sentences were the same. These 400 sentences will be referred to as the **sample** and they are the ones that will be used to categorize errors for each language pair.

In the previous section, by calculating POS metrics, we got a general idea of the kinds of errors that exist in the MT output. Some of the shortcomings of NMT in general are considered to be:

- long sentences
- unknown and rare words
- transliteration and compound words
- rich morphology
- out-of-domain data

All of these should be taken into account when categorizing MT output.

Regarding the language pairs, Swedish, Greek, and English all belong to the Indo-European family tree, with the difference that Greek belongs to an independent branch (Hellenic), while English belongs to the West-Germanic group of languages, and Swedish belongs to the North Germanic languages. Thus, one can surely expect some differences between these languages.

More specifically, from a purely personal point of view of the Swedish language in comparison to English, some of the differences present in the languages are the use of the auxiliary verb *do* to form questions in English, absence of the continuous tense in Swedish, as well as absence of verb inflections. Regarding tense again, Swedish, makes use of the present perfect, where English would require the past simple, and where English would use the auxiliary verbs *will* and *be going to*, to form the future tense, Swedish uses present simple. Regarding word order, even though they are both SVO languages, it is common in Swedish to invert subject and verb in a sentence. Finally regarding inflection, Swedish nouns and adjectives can be inflected to denote number (en väska (sing.) - väskor (pl)), definite (väskan (the bag), väskorna (the bags))

For the GtE language pair, the most obvious difference is inflection. English is a very lightly inflected language (for example, nouns are only inflected to mark plural), while Greek is a heavily inflected language, with inflections for nouns, pronouns, and adjectives including case, number, gender, and person. For verbs, there exist inflections for mood, voice, person, number, while also having inflection combinations for infinitives as well. Again, they both have a SVO word order, but due to the before mentioned aspects of the Greek language, the word order can be more free with VSO and other constructions allowed (e.g. *του Νίκου το σπίτι* which literally translates to *the Nick's the house*).

In the following parts we are going to manually categorize errors in the sentences, and then use them as comparison to the performance of an automatic error
detection process. For both parts the findings made by syntax oriented measures calculated in 5.2 will be taken into account, pointing to various possible mistakes in the MT output.

### 5.3.1. Manual Error Categorization

Manual error categorization and error analysis is a tedious and time consuming work, despite the fact that most of the time there is a reference translation available for guidance. This is due to the fact that more than one translations of the same sentence could be considered correct, and, as one can understand, this poses a lot of problems for this task.

We are going to use the following categories of errors for both the manual and the automatic categorization process (Popović and Ney, 2011):

- inflectional errors
- reordering errors
- missing words
- extra words
- incorrect lexical choices

Each category will be counted and summed up according to its appearance in the sample, and the results will be shown in 5.3.3 alongside the results of the automatic error categorization which is presented in the following section.

The procedure that is going to be followed here won’t be as strict as the one followed by automatic evaluation metrics. Rather it will be somewhere in the middle of strict and flexible, allowing for synonyms, word order changes, etc, as long as the meaning of the sentence is preserved. For example, an automatic evaluation metric would score low for the following sentences since the hypothesis does not exactly match the reference.

- **ref:** "Yesterday, I visited the farmers’ market to do my grocery shopping."
- **hyp:** "I went to the farmers’ market yesterday to buy some groceries."

The BLEU score acquired was only 0.19, even though the meaning they convey is the same. We, on the other hand, will allow for change in the placement of words (e.g. reordering error: ‘yesterday’) as long as they appear at their correct place following the syntactic rules of the language, as well as alternative expressions use (e.g. incorrect lexical choice: ‘buy groceries’ and ‘do the grocery shopping’, ’went’ and ‘visited’), as long as they retain the meaning of the sentence without altering it. What we will mark as errors though, are the words ‘my’ and ‘some’, as a missing and extra word respectively.

### 5.3.2. Automatic Error Categorization

As we saw in 5.2.4, out of all the syntax oriented metrics tested, the decomposition of WER over POS tags was one of the most informative ones, and the one that can potentially not only denote the quality of the MT output, but also help point out the errors in it. This is why it is going to be one of the main metrics used for automatic error categorization, which will be based on the approach by Popović and Ney (2011).
We have already introduced WER in 5.2.2, how it works and the logic behind it. Alongside WER, some variations of position independent word error (PER) will be used. PER is based on the same substitution, insertion, and deletion method as WER but it does not take into account the word order of the sentence, like WER does. The formula which is used for calculation is the following:

$$\text{PER} = \frac{1}{N_{\text{ref}}} \sum_{k=1}^{K} \min \{ d_{\text{PER}}(\text{ref}_{k}, \text{hyp}_{k}) \}$$

where

$$d_{\text{PER}}(\text{ref}_{k}, \text{hyp}_{k}) = \frac{1}{2} \left( |N_{\text{ref}_{k}} - N_{\text{hyp}_{k}}| + \sum_{e} |n(e, \text{ref}_{k}) - n(e, \text{hyp}_{k})| \right)$$

The counts of a word in the hypothesis and the reference sentence are used for the calculation, as visible.

By re-using the example in 5.2.2, slightly modified:

- **hypothesis**: "Mrs Commissioner, sometimes twenty-four hours is too much time."
- **reference**: "Mister Commissioner, twenty-four hours sometimes can be too much."

we get three PER errors, two substitutions (‘Mrs’, ‘Mister’, and ‘can’ ‘is’) and one deletion (‘be’).

Its downside though, is that it can never identify word order mistakes, nor can it identify which were substitution, insertion, and deletion errors. So what is going to be used in this thesis are the following metrics based on PER:

- **HPER** (hypothesis, precision-based PER)

$$\text{HPER} = \frac{1}{N_{\text{hyp}}} \sum_{k=1}^{K} \sum_{e} n(e, \text{herr}_{k})$$

- **RPER** (reference, recall-based PER)

$$\text{RPER} = \frac{1}{N_{\text{ref}}} \sum_{k=1}^{K} \sum_{e} n(e, \text{reerr}_{k})$$

- **FPER** (f-measure based PER)

$$\text{FPER} = \frac{1}{N_{\text{ref}} + N_{\text{hyp}}} \cdot \sum_{k=1}^{K} \sum_{e} \left( n(e, \text{reerr}_{k}) + n(e, \text{herr}_{k}) \right)$$

33
where $herr_k$ and $rerr_k$ are the number of words that do not appear in the reference and in the hypothesis respectively.

To explain, and taking once again the sentences used before as an example, the number of reference errors are 3 and the number of hypothesis errors are 2. The contribution to PER is 3, and if we normalize it over the length of the reference we get PER = 25%. RPER is then equal to 25% (3/12), HPER is 16.7% (2/12) and FPER is 21.7% ((2+3)/(11+12)). All these metrics can be used to determine how much each syntactic category takes part in the errors of the MT output, just as was done with WER.

These metrics are going to be used alongside WER to automatically categorize errors, as mentioned already, in the following way:

→ inflectional errors: RPER and the lemma of a word. If there are erroneous words in the hypothesis-reference pairs, whose lemma matches, then those are considered inflection errors.

→ reordering errors: WER plus RPER or HPER. For this we are going to first take all words which are present both in the hypothesis and the reference, and then check if they were also marked as a WER error. Those are considered reordering errors.

→ missing words: WER plus RPER on the lemmas. Errors present in the deletion list of the WER algorithm, which are also present as RPER error but do not share the same base form as any errors present as HPER errors. Those are considered missing words.

→ extra words: WER plus HPER on the lemmas. Errors present in the insertion list of the WER algorithm, which are also present as HPER error but do not share the same base form as any errors present as RPER errors. Those are considered extra words.

→ incorrect lexical choice: errors that do not fall in the category of either an inflectional or missing word errors. Those are considered incorrect lexical choices.

In the following section the results of these experiments are presented, compared, and discussed.

### 5.3.3. Results and Short Discussion

Table 5.6 presents the results acquired after manually categorizing and automatically categorizing the errors found in the sample for each language pair.

<table>
<thead>
<tr>
<th>Error/ Type</th>
<th>StE Manual</th>
<th>StE Automatic</th>
<th>GtE Manual</th>
<th>GtE Automatic</th>
</tr>
</thead>
<tbody>
<tr>
<td>inflectional errors</td>
<td>10</td>
<td>14</td>
<td>58</td>
<td>118</td>
</tr>
<tr>
<td>reordering errors</td>
<td>276</td>
<td>332</td>
<td>232</td>
<td>549</td>
</tr>
<tr>
<td>missing words</td>
<td>39</td>
<td>47</td>
<td>49</td>
<td>53</td>
</tr>
<tr>
<td>extra words</td>
<td>42</td>
<td>59</td>
<td>64</td>
<td>86</td>
</tr>
<tr>
<td>incorrect lexical choices</td>
<td>365</td>
<td>435</td>
<td>518</td>
<td>639</td>
</tr>
</tbody>
</table>

Table 5.6: Results per Language Pair

While going through the sentences manually, and categorizing them a number of observations could be made. Regarding the StE language pair, inflectional errors...
mainly affected verbs and nouns, and this was connected partly to reordering errors since many constructions were turned into the passive voice, thus, the changes in verbs. As far as missing words are concerned, the main categories were the words 'SEK' and 'old', while the rest of the instances were determiners, punctuation and words that added context meaning to the sentence, with the same thing visible in the extra words category. These results also reinforce the findings of 5.2.4 where determiners and punctuation have a relatively high percentage. Finally, regarding incorrect lexical choice errors, some patterns were detected which enabled to build a dictionary of common mistakes, and their equivalent correct term, which will assist us while formulating APE rules.

Some examples are given below, for each category:

→ inflectional errors
ref: "... the agency that decides prices and subsidies"
hyp: "... the authority that decides on price and subsidy."

→ reordering errors
ref: "... it is investigated further."
hyp: "... it is further investigated."

→ missing words
ref: "Avoid giving the child very sweet drinks ..."
hyp: "Avoid giving very sweet drinks ..."

→ extra words
ref: "The interpreter may participate on a speaker phone."
hyp: "The interpreter may participate in the meeting on a speaker phone."

→ incorrect lexical choice
ref: "... of medicines and stoma care products ..."
hyp: "... of medicines and ostomy items ..."

For the GtE language pair on the other hand, concrete conclusions could not be reached. Inflectional errors, once more, affected verbs and nouns more, for the same reason as the one mentioned for the StE language pair, but regarding the rest of the categories, no pattern could be detected. This can be attributed to the fact that the corpus consists of talks by various persons, and even though it belongs to the political domain, each person’s style is expressed through their speech so error patterns can not be identified and used for APE. Finally, it should be noted that specifically for this language pair, and while going through the examples manually, the reference translation seemed at times too free, to the point of distortion of the meaning of the original sentence, making the hypothesis sentence a better match.

Again, some examples are given below, for each category:

→ inflectional errors
ref: "... on their everyday life ..."
hyp: "... on their everyday lives ..."

→ reordering errors
ref: "The total investment cost ..."
hyp: "The total cost of investment ..."

→ missing words
ref: "... Mr Vatanen spoke to us of ...
hyp: "... Mr Vatanen spoke of ...

→ extra words
ref: "Mr President, Commissioner, as proof ..."

hyp: "Mr President, Commissioner, ladies and gentlemen, as proof ..."

→ incorrect lexical choice

ref: "... remarks concerning the key amendments."

hyp: "... remarks concerning the most important amendments."

In the previous paragraphs it was mentioned that sometimes the reference translation was too free, losing thus parts of the meaning of the original sentence. This is also obvious through the number of missing words, and extra word errors presented in the tables above, and it was present in both language pairs, being more prominent though in the GtE language pair.

Some examples are given below for reference.

→ GtE example 1:

• source: "Ζητώ να γίνει αποδεκτή εκείνη η τροπολογία της Επιτροπής Περιφερειακής Πολιτικής και Μεταφορών που εγκρίθηκε ομόφωνα και αφορά τον χρόνο εφαρμογής της οδηγίας.

• hypothesis: "I call for the amendment adopted by the Committee on Regional Policy and Transport, adopted unanimously, concerning the timing of the directive."

• reference: "The one unanimously adopted amendment of the Committee on Regional Policy and Transport, which concerns the timetable for implementing the directive, is something which I would urge you to support."

→ GtE example 2:

• source: "Είμαι στη διάθεσή σας."

• hypothesis: "I am at your disposal."

• reference: "I will keep myself free in this connection."

→ StE example 1:

• source: "I ett tidigt skede går det inte att skilja de två cancerformerna åt och man riskerar därför att behandlas för en cancerform man inte skulle ha fått några besvär av under sin livstid."

• hypothesis: "At an early stage, it is not possible to distinguish the two cancers and therefore you risk being treated for a cancer that you would not have had any problems during their lifetime."

• reference: "There is therefore a danger that you would be treated for a type of cancer that would never have caused any problems during your lifetime."

→ StE example 2:

• source: "En del mottagningar vill då att du skriver en så kallad egenanmälan, som ibland också kallas egenremiss eller egen vårdbegäran."

• hypothesis: "Some receptions then want you to write a so-called self-report, which is sometimes also called self-referral or your own care request."
• **reference**: "Some clinics want you to send them a letter first."

This kind of free style of translation in the reference can naturally lead to low scores when evaluating the hypothesis translation. This raises the issue of more than one translation needed to be used as reference, so as to ensure better performance, not tied to the style of only one translator.

Regarding the comparison between automatic and manual error categorization we can see that, generally, for both language pairs we can identify a difference in the numbers presented in table 5.6, which can be attributed to our flexible style versus the program’s strict style mentioned in 5.3, of our error categorization. There is a slight difference though in the variation between the two scores for the categories of missing and extra words, and incorrect lexical choices. These three categories are generally difficult to identify and tell apart, so this could be the reason behind those differences.

For reference a sentence was examined more closely, so as to validate the above hypothesis. The first and second ones are the reference and hypothesis sentence annotated manually, and the third and fourth ones are the same sentences annotated automatically.

**StE sentence:**

- **ref (man.):** "The interpreter may participate on a speakerphone."
- **hyp (man.):** "The interpreter can participate *in the meeting* *(extra words)* through speakerphone."
- **ref (aut.):** "The interpreter *may* *(incorrect lexical choice)* participate *on a* *(incorrect lexical choices)* speakerphone."
- **hyp (aut.):** "The interpreter can participate in the *meeting through* *(extra words)* speakerphone."

**GtE sentence:**

- **ref (man.):** "But the majority voted against *it* *(incorrect lexical choice)* ."
- **hyp (man.):** "However, the majority voted against this *corrigendum* *(extra words)* ."
- **ref (aut.):** "*But* *(incorrect lexical choice)* the majority voted against *it* *(incorrect lexical choice)* ."
- **hyp (aut.):** "However, the majority voted against this *corrigendum* *(extra words)* ."

As is visible the automatic error categorization script follows the reference strictly, that is why errors that were not considered as such during manual error categorization were marked. This explains the difference in numbers present in Tables 5.6 and 5.7.
5.3.4. Spearman’s \( \rho \) and Pearson’s \( r \) rank correlation coefficients

To be able to determine whether the automatic error categorization was meaningful and could potentially be useful, we must determine whether it correlated well with human judgment. To achieve this, Spearman’s \( \rho \) and Pearson’s \( r \) rank correlation coefficients were calculated for both language pairs. Both measures describe the relationship between variables using a monotonic function, but the difference between them is that Spearman’s \( \rho \) (Daniel, 1990) is used when the values come from an ordinal scale (e.g. a scale of satisfaction from 1 to 5) based on rank, while Pearson’s \( r \) (Rodgers and Nicewander, 1988) is used when the values come from an interval scale (e.g. temperature, age, etc.) assuming a linear relationship. Both of them vary from -1 (low correlation) to +1 (high correlation). Most of the times their scores are roughly the same for large data sets, but it is considered best to compute both, so as to extract useful information about the data in case differences are observed.

To calculate the scores, Scipy’s\(^8\) statistical functions package was used. Table 5.7 below shows the results obtained when these two metrics were implemented.

Two scatter plots, Figure 5.3 and Figure 5.4, are given as well, which represent the data scattered as points in a diagram. In a scatter plot, if the points appear in an increasing order (from bottom left to top right), it shows that the variables are correlated, with decreasing order meaning the opposite. They were visualised using the matplotlib\(^9\) library. Manual scores appear in axis X, while scores obtained from the automatic measure appear in axis Y.

<table>
<thead>
<tr>
<th>Language Pair/Values</th>
<th>Spearman’s ( \rho )</th>
<th>Pearson’s ( r )</th>
</tr>
</thead>
<tbody>
<tr>
<td>StE</td>
<td>0.99</td>
<td>0.99</td>
</tr>
<tr>
<td>GtE</td>
<td>0.89</td>
<td>0.91</td>
</tr>
</tbody>
</table>

**Table 5.7.:** Spearman’s \( \rho \) and Pearson’s \( r \) scores per Language Pair

![StE Scatter Plot](image)

![GtE Scatter Plot](image)

For the StE we can see that the automatic metrics are well correlated with human judgment, especially since both metrics scored the same. For the GtE, we also get a high correlation with human judgment, with the metrics not being identical, but nevertheless being high. The two scatter scores move from bottom

---

left to top right, having, thus, an increasing order, and reinforcing the fact that the variables are correlated. This strongly suggests that the automatic error detection method is meaningful and represent a valid automatic approach to the problem of error detection and error analysis.

5.3.5. Pipeline and HTML Visualization

At this stage all of the before implemented metrics were combined into a pipeline, so as to create a wholesome evaluation script that would enable an accurate depiction of the quality of a NMT output, through an informative output file. The output of an example can be found in the Appendix. (A)

Also in the Appendix, (B), one can find the output of a script which utilizes the incorrect lexical choices list from the error categorization step to produce an html file which includes the hypothesis sentence, with the incorrect lexical choice of the reference sentence given next to the equivalent hypothesis one in green. This can potentially be useful for the task of APE.
6. Automatic Post-Editing

As was mentioned in 2.4, APE is a vital part in any machine translation task, so as to ensure optimal quality of the output, and minimize the drawbacks of any machine translation system used. Since it is such a tedious work, any kind of automatization of it can be considered useful and less time costly.

The results in 5.3.3, and especially the comparison between a flexible manual categorization and a strict automatic one, show that the most important category that needs to be handled is the category of incorrect lexical choices, since these are the ones that could potentially denote domain specific errors in the vocabulary choice, or more appropriate ones that can improve the overall condition of the MT output. For this part regular expressions will be used to correct and transform the output of the sample, and then their impact will be assessed using different metrics, both for the sample, as well as the whole of the data sets.

6.1. GtE sample data

It should be noted that a significant effort was put into choosing Greek data that could potentially lead to meaningful results. New data were chosen twice, but both times error categorization was not able to reach clear conclusions as to what could be corrected. This can be attributed to two things. First of all, even though the corpus belonged to the political domain, the language it contained was spoken, so one can not expect clear, domain specific or not, patterns to arise from this kind of language. Even though it entailed speeches, political speech style vary a lot and may contain a lot of personal elements, to ensure persuasiveness. Secondly, as also mentioned again many times, the reference translation of the original sentences was, most of the time, too free to the point of meaning distortion, which led to an inability to identify error patterns that could be corrected, since at times, the hypothesis translation seemed more appropriate than the reference.

These are, thus, the reasons why the GtE language pair will not be handled in this section, and why, generally, data choice for APE rules is of extreme importance, to enable carrying out this task successfully. It is believed that a different corpus, one not spoken language based, would have provided more reliable data, since political speeches by various individuals are unlikely to produce patterns regarding the domain or the language itself.

6.2. StE sample data

For the StE language pair, though, more concrete conclusions could be reached. It was obvious after isolating the incorrect lexical choices, that they were medical terms that could not be correctly translated by NMT, making, thus, the output either completely erroneous, or too formal/not formal enough. Terms that were
encountered more than two times were taken into account to ensure frequency of appearance in the sample, as well as in the rest of the data set. Some examples are given below:

- **distortion of meaning**: get fluid / become dehydrated, winter jerky / winter vomiting disease

- **(in)formal choice of word**: ostomy items / stoma care products, antipyretic drug / temperature-reducing medicine

These kinds of words were the ones used in the APE rules. Apart from those some instances of missing words (e.g. ‘SEK’ or ‘old’) were also added in the correct context so as to ensure a more wholesome meaning.

First, common and syntactic metrics were calculated for the sample (as in sections 5.1, 5.2, so as to assess the quality of the translation before the APE rules. After that, the procedure used was to run the hypothesis sentences through a series of regular expressions, which aimed to transform or include the before mentioned. Finally, the same metrics were calculated again, as well as a re-run of the automatic error detection script, so as to check the impact of the rules. Then the same procedures were applied to the rest of the data to inspect whether the rules have an impact on the whole data set. The results are presented and discussed in the following section.

6.3. Results and Short Discussion

<table>
<thead>
<tr>
<th></th>
<th>Sample before APE</th>
<th>Sample after APE</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BLEU</strong></td>
<td>0.295</td>
<td>0.309</td>
</tr>
<tr>
<td><strong>METEOR</strong></td>
<td>0.321</td>
<td>0.327</td>
</tr>
<tr>
<td><strong>TER</strong></td>
<td>0.637</td>
<td>0.612</td>
</tr>
<tr>
<td><strong>POSBLEU</strong></td>
<td>0.632</td>
<td>0.642</td>
</tr>
<tr>
<td><strong>WPF</strong></td>
<td>6.610 %</td>
<td>6.699 %</td>
</tr>
</tbody>
</table>

Table 6.1.: Metrics before and after APE Rules, Sample

All three tables present in this section, show that the automatic correction script provides a better translation quality. The first three metrics in Table 6.1, being word based metrics, show that the changes in vocabulary result in better scores. BLEU and METEOR score higher than before APE, and TER shows that less PE effort is required with a smaller score. POSBLEU and WPF also score higher, which shows that the change in words, thus, in POS tags, match the reference better than the original hypothesis sentences.

Table 6.2 shows that nouns, verbs, proper nouns, and adjectives, the main categories present in the rules, contribute less to WER after APE. It should be noted that some categories like adjectives or auxiliaries show a slightly higher percentage. This could be attributed to a changed alignment of the sentences after the modifications made by the APE, which led to these differences, or to reordering ‘penalty’ by the WER algorithm.
Table 6.2.: WER Score of Various POS tags before and after APE Rules, Sample

<table>
<thead>
<tr>
<th>Tag</th>
<th>Sample before APE</th>
<th>Sample after APE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADJ</td>
<td>5.42%</td>
<td>5.78%</td>
</tr>
<tr>
<td>ADP</td>
<td>5.02%</td>
<td>5.02%</td>
</tr>
<tr>
<td>ADV</td>
<td>3.63%</td>
<td>3.60%</td>
</tr>
<tr>
<td>AUX</td>
<td>5.32%</td>
<td>5.38%</td>
</tr>
<tr>
<td>CCONJ</td>
<td>1.39%</td>
<td>1.32%</td>
</tr>
<tr>
<td>DET</td>
<td>5.22%</td>
<td>5.09%</td>
</tr>
<tr>
<td>INTJ</td>
<td>0.07%</td>
<td>0.07%</td>
</tr>
<tr>
<td>NOUN</td>
<td>18.73%</td>
<td>17.93%</td>
</tr>
<tr>
<td>NUM</td>
<td>0.26%</td>
<td>0.26%</td>
</tr>
<tr>
<td>PART</td>
<td>2.31%</td>
<td>2.31%</td>
</tr>
<tr>
<td>PRON</td>
<td>5.61%</td>
<td>5.61%</td>
</tr>
<tr>
<td>PROPN</td>
<td>1.12%</td>
<td>0.83%</td>
</tr>
<tr>
<td>PUNCT</td>
<td>8.12%</td>
<td>8.39%</td>
</tr>
<tr>
<td>SCONJ</td>
<td>1.12%</td>
<td>1.06%</td>
</tr>
<tr>
<td>SYM</td>
<td>0.00%</td>
<td>0.00%</td>
</tr>
<tr>
<td>VERB</td>
<td>8.49%</td>
<td>8.26%</td>
</tr>
<tr>
<td>X</td>
<td>0.40%</td>
<td>0.79%</td>
</tr>
</tbody>
</table>

Table 6.3.: Automatic Error detection before and after APE Rules, Sample

<table>
<thead>
<tr>
<th>Error Type</th>
<th>Sample before APE</th>
<th>Sample after APE</th>
</tr>
</thead>
<tbody>
<tr>
<td>inflectional errors</td>
<td>14</td>
<td>13</td>
</tr>
<tr>
<td>reordering errors</td>
<td>332</td>
<td>321</td>
</tr>
<tr>
<td>missing words</td>
<td>47</td>
<td>29</td>
</tr>
<tr>
<td>extra words</td>
<td>59</td>
<td>58</td>
</tr>
<tr>
<td>incorrect lexical choice</td>
<td>435</td>
<td>432</td>
</tr>
</tbody>
</table>

Finally, the results of the APE are even more clear in Table 6.3, where one can see that all error categories contain less examples after APE than before, which was to be expected since, for example the addition of the words 'SEK' and 'old' in the correct context minimized missing words (since they were only found in the reference), while a change of domain specific words to more appropriate ones affected all of the categories.

6.4. APE on the Rest of the Data

Seeing how these APE rules affected the sample in a positive way, the same rules were then applied to the rest of the sentences (RD) (hypothesis sentences minus the ones taken to be used as the sample), to check the impact of the rules, and detect whether the performance is representative for the whole data set. The three tables presented below show the results.

As one can see the results for RD differ to the ones presented in 6.3. Starting with table 6.4, we can see that almost all the metrics, word-based and syntax-based ones, show a drop in performance after the APE rules are applied to the
hypothesis sentences. This leads us to believe that neither the word replacements nor the word additions or the words’ POS tags of the APE matched the reference sentences, which led to this drop in the metrics.

For the other two tables though, table 6.5 and 6.6, the behaviour varies. Table 6.5 shows a slight improvement in percentage for almost all categories of POS tags for the WER metric. Adjectives, nouns, proper nouns, and verbs which are the main elements of the APE rules do show an improvement, and seeing how WER does not account for reordering, we could argue that some of the changes

<table>
<thead>
<tr>
<th>Table 6.4.: Metrics before and after APE Rules, RD</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>RD before APE</strong></td>
</tr>
<tr>
<td><strong>BLEU</strong></td>
</tr>
<tr>
<td><strong>METEOR</strong></td>
</tr>
<tr>
<td><strong>TER</strong></td>
</tr>
<tr>
<td><strong>POSBLEU</strong></td>
</tr>
<tr>
<td><strong>WPF</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 6.5.: WER Score of Various POS tags before and after APE Rules, RD</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>RD before APE</strong></td>
</tr>
<tr>
<td><strong>ADJ</strong></td>
</tr>
<tr>
<td><strong>ADP</strong></td>
</tr>
<tr>
<td><strong>ADV</strong></td>
</tr>
<tr>
<td><strong>AUX</strong></td>
</tr>
<tr>
<td><strong>CCONJ</strong></td>
</tr>
<tr>
<td><strong>DET</strong></td>
</tr>
<tr>
<td><strong>INTJ</strong></td>
</tr>
<tr>
<td><strong>NOUN</strong></td>
</tr>
<tr>
<td><strong>NUM</strong></td>
</tr>
<tr>
<td><strong>PART</strong></td>
</tr>
<tr>
<td><strong>PRON</strong></td>
</tr>
<tr>
<td><strong>PROPN</strong></td>
</tr>
<tr>
<td><strong>PUNCT</strong></td>
</tr>
<tr>
<td><strong>SCONJ</strong></td>
</tr>
<tr>
<td><strong>SYM</strong></td>
</tr>
<tr>
<td><strong>VERB</strong></td>
</tr>
<tr>
<td><strong>X</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 6.6.: Automatic Error detection before and after APE Rules, RD</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>inflectional errors</strong></td>
</tr>
<tr>
<td><strong>reordering errors</strong></td>
</tr>
<tr>
<td><strong>missing words</strong></td>
</tr>
<tr>
<td><strong>extra words</strong></td>
</tr>
<tr>
<td><strong>incorrect lexical choice</strong></td>
</tr>
<tr>
<td><strong>X</strong></td>
</tr>
</tbody>
</table>
made were actually correct. Table 6.6 shows a drop in inflectional and reordering errors, but an increase for the rest of the categories, which were the ones handled by the rules, and especially for the category of incorrect lexical choice as can be seen.

To investigate this puzzling behaviour, another sample of 150 sentences was taken from RD, so as to aid in determining the reason behind the low impact of the APE rules. What was detected was an inconsistency in the reference sentences, in relation to the 200 sentences sample, which explains why the APE rules failed to improve on the rest of the data set.

One type of inconsistency was the placement of the word ‘SEK’. In the sample’s reference sentences the word was always before the digits, which led to the formulation of a specific regular expression that would add this word in the correct place, since it was missing in the hypothesis sentences. In RD, this word’s placement varied from being placed after the digits, or missing altogether. Another example was the sign of the Celsius temperature that was at times ‘° C’ while in other cases it was ‘degrees Celsius’.

Another type of inconsistency detected was in the word choices or their spelling. The word ‘mucous’ is generally spelled as such when the word ‘membrane’ follows, differentiating it from the word mucus. In the reference there were many cases where the spelling was mixed up, not following this rule. The same could be said for the word ‘temperature’ and ‘fever’. In the sample’s reference sentences the word ‘fever’ was not used, with the word ‘temperature’ taking its place. In the RD an equal use of the words was detected.

Finally, it was also found that some of the APE rules’ words were not present in the RD, which means that the sample was not as representative of the whole data set.

All of these prove that more than one reference sentences are needed to counteract for such cases when the translator was not consistent in her translation, or for cases where more than one translators worked on the reference sentences. Lack of inconsistency can lead to low APE results, precisely since the automatic aspect of the the task requires consistency to work.
In this work a number of processes regarding machine translation output were undertaken, for two different language pairs, Swedish to English (1177.se website), and Greek to English (a session of the Europarl corpus). We initialized the study by measuring the quality of the neural machine translation output for both language pairs using commonly used metrics, namely BLEU, METEOR, and TER. All three metrics denoted a bad translation quality with Swedish to English language pair scoring lower for BLEU and METEOR compared to Greek to English, but higher in TER.

Since these metrics are not able to point out issues and errors of the translation, other metrics were then calculated, based on syntax. The output and the reference sentences were passed through NLP cube to obtain the POS-tags along with syntax specific information, so as to be used by the metrics. POSBLEU followed by WPF and WER of various POS-tags were calculated. The first showed a good translation quality, differing in BLEU by 0.4, while the second showed a rather disappointing result. Since the first is purely syntax based, while the second combines words and their POS-tags, we inferred that the hypothesis sentences differed from the reference ones in words, but not syntactically, which lead us to believe that this meant instances of synonyms, and use of alternative expressions use. The decomposition of WER over POS-tags also reinforced this by showing that the four categories that played the most important role in the errors between the sentences were nouns, adjectives, pronouns, and verbs.

Having these findings in mind, we then moved on to error categorization, manual and automatic, for the following five error categories:

→ inflectional errors
→ reordering errors
→ missing words
→ extra words
→ incorrect lexical choice

A sample of 200 sentences were taken from each language pair to aid in the manual error categorization, which was done in a more flexible and forgiving way, compared to the automatic error categorization, which is rather strict. For the automatic error categorization a combination of WER and PER errors for the hypothesis and reference sentences were combined. Spearman’s $\rho$ and Pearson’s $r$ were calculated, and proved that the automatic error categorization script results were well correlated to human judgment.

The error categorization step revealed that the most important errors present in the machine translation output were incorrect lexical choices and important words missing from the reference, and these were the ones that were decided to be handled by automatic error correction.

For the Greek to English language pair automatic post editing was not possible. The reference sentences were examples of an extremely free translation of the
original sentence, to the point of meaning distortion. This in combination with the fact that the spoken language present in the sentences lacked a clear pattern of errors, led to an inability of handling them at that step.

For the Swedish to English pair, rules were then crafted based on regular expressions, which altered the hypothesis sentences. The corrected sentences of the sample were then compared to the reference sentences, and proved to have a positive effect on them, with all metrics showing an improvement.

Seeing how the rules had a positive impact on the sample, they were also applied to the rest of the data. The metrics which were then calculated showed a decrease in most of the metrics, which was puzzling. Another sample was then taken so as to detect the reason behind the failure of the rules. What was found was that not only was the sample not representative of the whole data set, but also showed a continuous inconsistency in the reference sentences, where the missing words handled by the rules were placed at different spots, while for the incorrect lexical choices, the words used were often changed, with no concrete reason being visible as to why.

For machine translation handling, thus, more than one references is necessary so as to ensure a wholesome comparison and wholesome results, while also making sure that the reference translations are consistent, so as to enable correct and effective automatic post editing. It must be mentioned here, though, that multiple references for the same source text are generally hard to come by, since they are not only time and resource consuming, but also not available for, for example, low-resource languages.

This study can be enriched by incorporating many other methods and metrics. Apart from the three common metrics used here, other, heavily used ones, could also be calculated, such as NIST, LEPOR, or ROUGE, which are metrics also well correlated to human judgment. NIST (Doddington, 2002) is a metric which was based on BLEU, but calculates the quality of machine translation output by assigning particular weights to n-grams that are more rare. LEPOR (Han et al., 2012) is derived from a combination of many different metrics used in the field, for example precision, recall, n-gram penalty, and others, while the different ROUGE (Lin, 2004) metrics work by also comparing hypothesis sentence to a number of references.

Other metrics based on syntax could be used as well, Liu and Gildea (2005) proposed the use of syntactic trees, constituent labels, and head-modifier dependencies, to check the fluency of the hypothesis sentence compared to the reference. It was shown that most of their proposed metrics outperformed BLEU. For these to be calculated, appropriate parsers that perform well are needed for the target and source languages. Use of dependency structures, output by a weighted constraints dependency grammar parser was proposed by Duma et al. (2013), as a syntax oriented metric, also well correlated to human judgment for the language pairs it was tested on.

Regarding the error detection step, there is a wide variety of errors that can also be targeted, apart from those examined in this paper. Tezcan et al. (2016b) propose the use of dependency parsing and treebank querying to detect and target grammatical errors, and testing showed that its performance is high. Goto and Tanaka (2017) tried to tackle the problem of untranslated content in neural machine translation by using cumulative attention probability and back translation.
probability, showing that each one worked well to detect untranslated content, while a combination of both lead to more improvements. The problem of cross-lingual semantic divergence present in neural machine translation output was investigated by Carpuat et al. (2017), by using a cross-lingual textual entailment system to filter out these kinds of sentences from training data, something which led to an enhancement in the quality of the output.

Finally, regarding automatic post-editing, other methods which do not employ regular expressions could be used to compare their performance with the performance of the method used here. Vu and Haffari (2018) describe a completely automatic procedure based on a neural programmer-interpreter approach where the programmer component performs edit actions, and the interpreter component executes them, but also keeps track of the actions already generated before deciding on the next one. Their results point to a +1 BLEU score and -0.7 TER scores, for the German-English language pair, outperforming previous neural models for PE. Negri et al. (2018) also propose a completely automatic post editing online system, which is based on a trained model, and which takes user input into account to improve the system’s performance, with positive results in the generic but not the specialized translations.
A. Pipeline

Example output of evaluating the candidate "Mrs Commissioner, sometimes twenty-four hours is too much time." against the reference "Mister Commissioner, twenty-four hours sometimes can be too much time." The common metrics, followed by syntax oriented ones, and finally an error categorization, are visible.

<table>
<thead>
<tr>
<th>POS-tag</th>
<th>WER percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADJ</td>
<td>0.00%</td>
</tr>
<tr>
<td>ADV</td>
<td>20.00%</td>
</tr>
<tr>
<td>AUX</td>
<td>20.00%</td>
</tr>
<tr>
<td>CCONJ</td>
<td>0.00%</td>
</tr>
<tr>
<td>DET</td>
<td>0.00%</td>
</tr>
<tr>
<td>INTJ</td>
<td>0.00%</td>
</tr>
<tr>
<td>NOUN</td>
<td>20.00%</td>
</tr>
<tr>
<td>NUM</td>
<td>20.00%</td>
</tr>
<tr>
<td>PART</td>
<td>0.00%</td>
</tr>
<tr>
<td>PRON</td>
<td>0.00%</td>
</tr>
<tr>
<td>PUNCT</td>
<td>30.00%</td>
</tr>
<tr>
<td>SCONJ</td>
<td>0.00%</td>
</tr>
<tr>
<td>SYM</td>
<td>0.00%</td>
</tr>
<tr>
<td>VERB</td>
<td>0.00%</td>
</tr>
<tr>
<td>X</td>
<td>0.00%</td>
</tr>
</tbody>
</table>

The following table shows how the different POS tags contribute to the WER score.

<table>
<thead>
<tr>
<th>Types of error</th>
<th>Number of errors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inflectional errors</td>
<td>1</td>
</tr>
<tr>
<td>Reordering errors</td>
<td>1</td>
</tr>
<tr>
<td>Missing words</td>
<td>1</td>
</tr>
<tr>
<td>Extra words</td>
<td>0</td>
</tr>
<tr>
<td>Incorrect lexical choices</td>
<td>1</td>
</tr>
</tbody>
</table>
B. HTML

Example of the html visualization of the hypothesis "Any linear reduction in subsidies is unreasonable and usually unfair, that is certain." against the reference "A linear reduction in aid is unimaginative and usually unfair, that is apparent.". Words that are considered incorrect lexical choices are given in green.

Any (A) linear reduction in subsidies (aid) is unreasonable (unimaginative) and usually unfair, that is certain. (apparent.)
Bibliography

Bentivogli, Luisa, Arianna Bisazza, Mauro Cettolo, and Marcello Federico (2016). “Neural versus Phrase-Based Machine Translation Quality: a Case Study”. In: EMNLP.


Popović, Maja, Adrià de Gispert, Deepa Gupta, Patrik Lambert, Hermann Ney, José B. Mariño, Marcello Federico, and Rafael E. Banchs (2006). “Morphosyntactic Information for Automatic Error Analysis of Statistical Machine Translation Output”. In: WMT@HLT-NAACL.


Sun, Yanli (2010). “Mining the Correlation between Human and Automatic Evaluation at Sentence Level”. In: LREC.


