Statistical Post-editing and Quality Estimation for Machine Translation Systems

Hanna Bechara

B.Sc., B.A.

A dissertation submitted in fulfilment of the requirements for the award of

Masters by Research (M.Sc.)

to the

Dublin City University
School of Computing

Supervisors: Josef van Genabith, Raphael Rubino, Yifan He and Yanjun Ma

July 2013

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of M.Sc.. is entirely my own work, that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge breach any law of copyright, and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Signed:

(Candidate) ID No.: 59119837

Date:

Contents

List of Figures

List of Tables

Abstract

Statistical post-editing (SPE) has been successfully applied to RBMT systems and, to a less successful extent, to some SMT systems. This thesis investigates the impact of SPE on SMT systems. We apply SPE to an SMT system using a new context-modelling approach to preserve some aspects of source information in the second stage translation. This technique yields mixed results, but fails to consistently improve the output over the baseline. Furthermore, we compared the results to those of an RBMT+SPE system and a pure SMT system, using both automatic and human evaluation methods. Results show that while automatic evaluation metrics favour a pure SMT system, manual evaluators prefer the output provided by the combined RBMT+SPE system. We investigate the use machine learning methods to predict which sentences would benefit from post-editing, however, as the oracle score for both SMT and SMT+SPE was not much higher than the two systems alone, we decided to compare two systems that had a higher upper bound. Combining our analysis with machine learning techniques for quality estimation, we are able to improve the overall output by automatically selecting the best sentences from each of the SMT and RBMT+SPE systems.

Chapter 1

Introduction

1.1 Machine Translation and Post-editing

Machine translation has become a commercial reality, based on over 50 years of research and study[1]. While machine translation has not reached the point where it can fully replace human translators, it has successful been integrated into the localisation[2] process, cutting down the amount of cost and labour required. The first real attempts at machine translation did not happen until the 1950's, and were based around ideas of information theory, code-breaking, and the underlying principles of natural language. Some early systems used large bilingual dictionaries and hard-coded rules, much like the rule-based systems of today. While early research gained momentum and funding, the real problem of machine translation turned out to be a lot more complicated than originally assumed. Machine translation was found to be substantially less accurate than human translators, and due to the limited technology at the time, less efficient and more expensive. However, research picked up again in the 1980's concentrating on primarily rule-based systems. Rule-based systems rely on hand-crafted rules for specific language pairs that are based on each of the language's individual characteristics.

[1]The overview presented in this section is based on the overview provided by (Specia et al., 2009).

[2]Localisation is the industrial process of adapting digital content to culture, locale and linguistic environment.

Rule-based systems still make up a significant part of commercial machine translation systems today, whether they are commercial systems such as Systran[1], or open-source systems like Apertium[2].

As language resources, and especially parallel language corpora, became more readily available, research began to take a more data-driven approach to machine translation. The quick advancement of computing power made processing and training models on large texts viable and without great cost. The 90s introduced the first statistical model, developed by IBM. The system made minimal use of dictionaries and linguistic rules, and instead treated translation as a decoding problem. It made use of parallel corpora to address this problem, and was found to produce more natural and less literal translations than previous rule based systems.

Recent advancements in the field of machine translation and improvements in MT quality have led implementations of various MT systems to be considered for commercial purposes. However, as machine translation is less accurate and less fluent than professional human translation, it is often used commercially in combination with post-editing. Post-editing is the process of "correcting" machine translation output, and is traditionally carried out by a person referred to as a "post-editor". In situations where MT output is already of fair to good quality, this process can be faster than full manual translation, and is rapidly becoming a viable alternative with the increased demand for localisation worldwide.

Post-edited output is of higher quality but naturally more time-consuming and expensive than raw MT output. Several toolkits to facilitate human post-editing have been released along with machine translation tools and studies into post-editing difficulty and adequacy have been carried out by researchers in the field of machine translation.

As human post-editors are still a costly resource, researchers have searched for ways to automate this error-correcting step, developing automatic post-editors.

[1] www.systransoft.com
[2] www.apertium.org

Automatic post-editors serve the same function as human post-editors; they correct errors in the machine translation output, targetting a wide variety of errors from lexical choice to grammatical agreement. Like human post-editors, automatic post-editors might use information unavailable to the translation system, and usually focus on correcting a specific type of error. Statistical post-editing is a type of post-editing system that uses a statistical machine translation system to perform a mono-lingual translation of the output of the machine translation system. The intuition is that this second translation will improve the output by introducing new information, such as new parallel data, lexical information, or domain-specific data, into the system.

Statistical post-editing of the output of RBMT and SMT systems is an active field of research and RBMT + SPE pipelines are by now a commercial reality, and available, for example, in more recent releases of Systran's machine translation system. Statistical post-editing of rule-based machine translation systems (Simard et al., 2007; Terumasa, 2007; Kuhn et al., 2010) has shown (in some cases) spectacular improvements in translation quality measured in terms of automatic evaluation scores. Furthermore, SPE has also been applied to the output of statistical MT (SMT) systems (Oflazer and El-Khalout, 2007; Potet et al., 2011; **?**; Rubino et al., 2012), albeit with more mixed results. The question of why SPE fails to improve on SMT as much as it improves on RBMT is an interesting one. However, to date, despite considerable interest in the area, the comparison between SPE pipelines and pure SMT and RBMT systems is not fully researched.

1.2 Research Questions

Our initial objective is to investigate in more detail whether and to what extent state-of-the-art PBSMT technology can be used to post-edit itself, i.e. its own output. We also extended our research to different machine translation systems, such as Rule-based Machine Translation systems. We capture our objectives in

terms of the following research questions, which are expanded on in the section following the research question:

- Can a monolingual second-stage SMT system improve the translation quality of the output, in particular that of a first stage SMT system?

Automatic post-editing corrects errors by introducing new information previously unavailable to the decoder. This is also true for the statistical post-editing of rule-based systems. Previous research in this field has shown that an SMT system trained on manually corrected output of rule-based systems can be used to improve the output (Simard et al., 2007; Terumasa, 2007; Kuhn et al., 2010), or even tune it to a specific domain (Isabelle et al., 2007; Rubino et al., 2012). However, when applied to an SMT system, the statistical post-editing system is not adding that much new information. The same engine is being re-used, with the first-stage system trained on the source and target language bitext data, and the second stage post-editing system trained on target-side mono-lingual bitext data consisting of first-stage system output and pristine reference data. In this case, the answer to the question of whether statistical post-editing can be used to correct its own output is much less clear-cut, and remains an open question in the field. In an attempt to answer this question, we apply a statistical post-editing system to the output of a statistical phrase-based machine translation system, using a ten-fold cross-validation technique to create our new training sets and context-modelling approach to the preserve some of the original source information in the second-stage system.

- How do SMT, RBMT + SPE and SMT + SPE compare when the statistical models are trained on the same data?

Previous work has shown that SPE, when applied to RBMT, can improve the overall output's quality over the RBMT system on its own. However, how this new output compares to the performance of the SMT system on its own is a question

that has not been fully investigated. The question of whether or not using the two systems in combination produces a better, more accurate, output than each of the systems on its own is a question we seek to answer as part of this research. To supplement automatic evaluation scores, we also use human evaluators to manually rank the output of each of these systems, and perform an in-depth study of the errors introduced, and corrected, by the post-editing system.

- Can we use quality estimation techniques to combine the best sentences outputted by the different post-editing and baseline systems?

The statistical post-editing system both corrects and introduces new errors into the machine translated output. While some sentences improve after post-editing, others degrade in quality either by diverging too far away from the source, or due to new errors introduced by the post-editor. We investigate whether or not we can use quality estimation techniques to choose the best sentences from both the post-edited and baseline systems.

1.3 Roadmap

The remainder of this thesis takes on the tasks in an attempt to answer these questions. We provide the necessary background work that frames the context of our own experiments and contributions. The following paragraphs describe the organisation and structure of this thesis.

Chapter 2 reviews previous research into statistical post-editing and our motivation for this study. We introduce the tools and techniques used throughout our research, and provide a snapshot of the history of statistical and rule based machine translation systems. We further outline the evolution of statistical machine translation which has led to the current state-of-the-art phrase based models used in our study. Furthermore, we introduce the concept of statistical post-editing and review some of the previous work in the field. Finally, we provide

an overview of quality estimation for machine translation output, both with and without reference translations, and the relevant research to date.

Chapter 3 explores the use of statistical machine translation methods to post-edit the output of a rule based and a statistical machine translation system, and details a novel word-alignment-modelling approach designed to preserve source context information. In this chapter we present our post-editing system and report results observed from applying it to both a rule-based (Systran) and a phrase-based statistical machine translation system (Moses (Koehn et al., 2007)). Additionally, this chapter explores a novel context-modelling approach to SPE which preserves the source context information in the post-editing phase. Experiments were conducted on a Translation Memory and a publicly available corpus. Overall our results were mixed: while we were able to achieve some improvements on the Translation Memory using the context-modelling approach, we were unable to replicate these results with a publicly available corpus. This prompted us to reinvestigate the data splits (training, tuning and testing) in the original Translation Memory data based experiments, showing a bias in the selection of the test data.

Chapter 4 compares these methods and delves deeper into the errors corrected by each of the post-editing systems. We use both manual and automatic evaluation techniques to compare the output of each system both before and after post-editing. We also conduct a detailed analysis of the errors introduced and corrected by the statistical post-editing system.

Chapter 5 looks at ways to classify sentences from the raw SMT output and the post-edited RBMT output, and select the better sentences to create the optimal output, using machine learning techniques previously used for quality estimation tasks, and prediction of post-editing effort. In this chapter, we document our work on training a classifier to select the better system's output on a sentence level, based on a number of system-independent features which we extracted based on our analyses of errors in Chapter 4. In addition to a classification model, we

trained a regression model that extends the previous approach by estimating continuous scores such as BLEU or TER at the sentence level, allowing us to combine the output of different MT systems by selecting the output that is predicted to be the best given the input.
Chapter 6 draws overall conclusions from previous chapters, and presents directions for possible future research.

1.4 Publications

Two papers were published in peer-reviewed conference proceedings as part of the research presented in this thesis. Our design and implementation of a statistical post-editing system for a statistical machine translation system and the source context-aware approach modelled for it were presented in (**?**). The results of our evaluation of the SPE system applied to different machine translation output, along with an analysis of the performance of the SPE system, are presented in (Béchara et al., 2012).

Publications from Thesis

- 2011 - H. Béchara, Y. Ma and J. van Genabith. *Statistical Post-editing for a Statistical MT System.* In Proceedings of MT Summit 2011, Xiamen, China.

- 2012 - H. Béchara, R. Rubino, Y. He, Y.J. Ma, J. van Genabith. *An Evaluation of Statistical Post-editing Systems applied to RBMT and SMT Systems*, Conference on Computational Linguistics (COLING), Mumbai, India

Chapter 2

Machine Translation, Post-Editing, and Quality Estimation

Of the several different approaches that have been implemented for machine translation over the years, we have focused on rule-based (RBMT) and statistical (SMT) approaches. In this section we briefly introduce both paradigms and outline the strengths and shortcomings of each. We review background research in statistical post-editing and quality estimation. We highlight previous work in combining RBMT, SPE and Quality Estimation, and show where our work fits in with the already existing research.

In this chapter we will cover:

- Statistical and rule-based machine translation and their respective strengths and weaknesses
- Manual and automatic post-editing, and in particular statistical post-editing
- Quality estimation and machine learning as a method to estimate translation quality

2.1 Rule-Based Machine Translation

Rule-based systems construct translations by using usually hand-crafted linguistic rules and bilingual dictionaries for a given language pair (Lagarda et al., 2009). Wide-coverage systems rely on large-scale lexical and morphological, semantic, and syntactic information. In the early days of machine translation, rule based machine translation was the dominant paradigm. Even today many of the commercially available and successful machine translation systems are rule-based. These systems work by analysing sentences in the source language (SL) morphologically, syntactically and semantically, and then structurally convert the sentences into the target language (TL) using the rules and the dictionary. The dictionaries themselves are often far more sophisticated than human dictionaries, as they need to reflect the grammatical properties of the words.
Rule-based systems have the advantage in that they tend to provide grammatically fluent and predictable quality translations. They build translations with well-formedness and grammatical correctness in mind. On the other hand, they are not easy to adapt to new expressions and different domains. Because RBMT usually looks for exact matching rules, it may fail to produce a complete translation when it cannot find rules that match part of the input.

2.2 Statistical Machine Translation

Since the introduction of statistical machine translation in 1990 Brown et al. (1990), data-driven research, in particular SMT, has become the most dominant strand of research in machine translation, superseding even rule-based systems. In contrast with rule-based systems, statistical machine translation systems build statistical models based on the analysis of existing parallel corpora. While statistical systems can produce unpredictable and in some cases somewhat erratic results, they tend to be more robust than rule-based systems (Thurmair, 2004).

2.2.1 The Noisy Channel Model

Statistical methods operate on the basis that every sentence in the source language has a determinable probability to be a translation for a given sentence in the target language (Wang, 1998). The Noisy Channel Model treats translation as a decoding problem, where the source sentence is nothing more than a scrambled distortion of the target translation. The challenge is finding the target sentence with the maximum probability of being the correct translation. This probability is modelled by Bayes' theorem (Equation 2.1):

$$P(\mathbf{T}|\mathbf{S}) = P(\mathbf{T})\frac{P(\mathbf{S}|\mathbf{T})}{P(\mathbf{S})} \tag{2.1}$$

where **S** is the text in the source language and **T** is the text in the language we are translating into. Therefore, our aim is to find a **T** which maximises $P(\mathbf{T}|\mathbf{S})$.
Given that P(**S**) is a given constant, the best translation $\hat{T}$ can therefore be captured in equation (2.2)

$$\hat{T} = \underset{T}{\operatorname{argmax}}\, P(\mathbf{T}) \cdot P(\mathbf{S}|\mathbf{T}) \tag{2.2}$$

Equation (2.2) divides our translation problem into two sub-problems, which can be addressed through two different SMT components: the language model and the translation model.
The language model, represented by P(**T**) in equation (2.2), does not reflect on the input text, but captures the well-formedness (as a string) of the target sentence itself. Therefore, the language model focuses more on fluency, given a particular language. SMT systems, in general, use n-gram, or word sequences instead of full sentences for language models. This is because it is highly unlikely to find the exact sentence, even in a very large corpus. The language model calculates the probability of n-grams based on the probabilities of individual words. The probability of each word is calculated based on the words preceding it, as shown in 2.3. As such, the language model aims to assign higher probabilities to sentences

that are better formed and syntactically correct.

$$P(T) = P(T_1)P(T_2|T_1)...P(T_n|T_1T_2...T_{n-1}) \tag{2.3}$$

where T is an n-gram made up of a sequence of n words ($T_1 to T_n$). In order to avoid the problem of zero probabilities for unseen n-grams, a smoothing technique is generally employed. The generally preferred smoothing method is the Modified Kneser-Ney smoothing proposed in (Kneser and Ney, 1995).

The translation model $P(S|T)$ searches for the sentence that best represents the intent and meaning of the source sentence. The translation model studies examples of translations from **S** to **T** in order to assess the adequacy of the translation. These examples are aligned either on a word-to-word basis, or on a phrase-by-phrase basis, and given a probability estimate. In the alignment step, source words are mapped to target words between parallel sentences. Alignment algorithms vary depending on whether the model is phrase-based or word-based, and on which model is being used.

In simple cases, this will be a 1 to 1 alignment, meaning each source word can only be mapped to a single target word. These mappings, denoted as α are also assigned probabilities based on equation 2.4.

$$\alpha = \underset{\alpha}{\operatorname{argmax}}\, P(\alpha|S,T) \tag{2.4}$$

Current state-of-the-art SMT systems rely on phrase-based models, which are be further discussed in Section 2.2.2.

2.2.2 Phrase-Based Models

While early SMT systems relied on word-to-word translation, current approaches use phrases as the basic unit of translation. A phrase can be defined as a contiguous sequence of words. However, in SMT, these sequences are not necessarily linguistically motivated. By using phrases instead of words, the

translation model can take into account the local context of the words being translated.

Phrase-based translation models map a sequence of words in the source language to a sequence of words in the target language. These phrases need not be identical in length.

In phrase-based models, $P(\mathbf{S}|\mathbf{T})$ in equation (2.4) is further decomposed into a phrasal and a re-ordering model. $P(\mathbf{S}|\mathbf{T})$ is defined in equation (2.5).

$$P(\bar{S}_1^I|\bar{T}_1^I) = \Pi_{i=1}^I \phi(\bar{S}_1^I|\bar{T}_1^I) d(start_i - end_{i-1}) \tag{2.5}$$

where $\bar{S}_I$ represents a source phrase, and $\bar{T}_I$ represents its target translation. $\phi(\bar{T}_1^I|\bar{S}_1^I)$ represents the phrasal translation model and $d(start_i - end_i - 1)$ represents the position-based re-ordering model.

The phrasal translation model is the probability of each phrase, based on a phrase translation table extracted from a symmetrically aligned corpus. These phrase pairs are obtained by applying a set of heuristics, which extracts them based on the constraint that they must be consistent with the word alignment. The word alignments are extracted bi-directionally, and the intersection of the two alignments yields the high-precision alignment to identify consistent phrase pairs. Once the phrases are extracted, their probability is estimated using Maximum Likelihood Estimation, as demonstrated in equation (2.6).

$$\phi(\bar{S}|\bar{T}) = \frac{count(\bar{T}|\bar{S})}{\Sigma_{S_i} count(\bar{T}|\bar{S})} \tag{2.6}$$

The re-ordering distance is estimated based on a pre-defined model, a decay function defined in equation (2.7).

$$d(x) = \alpha^{|start_i - end_{i-1}|} \tag{2.7}$$

where the parameter $\alpha \in [0, 1]$.

Combining the translation model with the language model P(T), which ensures

that the output is fluent in addition to adequate, produces a basic phrase-based SMT model, as defined in equation (2.8).

$$\underset{T}{\operatorname{argmax}}\, P(T|S) = \underset{T}{\operatorname{argmax}}\, \Pi_{i=1}^{I} \phi(\bar{S_1^I}|\bar{T_1^I}) d(start_i - end_{i-1}) Pi_{i=1}^{|T|}(T_i|T_1 ... Te_{i-1}) \quad (2.8)$$

2.2.3 Log-Linear Models

The standard phrase-based model was later extended to the log-linear model described in equation (2.9), which allows the integration of additional translation features in estimating the translation model (Och and Ney, 2003). These feature functions (h_m) are derived from the translation or language model of the SMT system and are assigned different weights (λ_m) based on their importance.

$$\hat{T} = \underset{T}{\operatorname{argmax}}\, exp \sum_{m=1}^{M} \lambda_m h_m(\mathbf{S}, \mathbf{T}) \quad (2.9)$$

From equation (2.9), we can infer the steps or sub-problems that statistical machine translation will have to solve. First, we need to solve the modelling problem, and build a model from which we can derive our feature functions (h_m) and provide a framework to calculate P($\mathbf{S}$) and $P(\mathbf{T}|\mathbf{S})$. The second problem is that of tuning, where the weights (λ_m)are determined. And finally, we have the problem of decoding, where, given the translation model and fully defined parameters, we can most efficiently identify a target sentence T for a given source sentence S.

2.2.4 Decoding

The actual decryption of new text into the target language is the job of the decoder. The decoder finds the best scoring translation, based on the reordering model probability, the language model probability, and the phrase translation probability discussed in Section 2.2.2. As an exhaustive search would be too costly,

the decoder implements heuristic search methods to find the best translation, or one as close to the best as possible.

The decoder starts by segmenting the input text into all possible phrases. It then collects all possible *translation options* based on the applicable alignment phrases. The decoder then generates the target hypotheses from left to right, by creating a new set of branches each time a new phrase is expanded into its possible target phrases. The text has no branches to begin with, and its "cost" is set to 1. Every time the decoder branches off, the cost of this new branch is calculated based on the cost of the previous branch and the relevant features of the new phrasal translation (which include the translation, reordering, and language model). Once the final state has been reached, that is all source words have been covered, the path with the lowest cost is identified as the sentence with the highest overall probability.

As the search space can potentially grow quite large, many decoders employ strategies that prune out hypothesis that fall outside a certain threshold. A common way to do this is by enlisting the use of hypothesis stacks organised by the number of words in the sentence that have been translated. Whenever a new hypothesis is generated, it is added to its relevant stack. If the size of a stack grows beyond a certain predefined limit, it is pruned and only the top scoring hypotheses are retained. The best translation is then chosen from the stack that covers all foreign words. Additionally, the decoder needs to take into account the estimated future cost of a hypothesis, to avoid bias towards translating the easy part of the sentence first. The future cost is estimated based on the untranslated sequence's language model and translation model.

2.3 Statistical Post-editing

It is common practise to post-edit automatic and semi-automatic translation outputs in order to correct the errors present in the MT output. This post-editing

is usually carried out by human post-editors and translators. While the use of machine translation systems and post-editors cuts down translation time significantly, human evaluators remain a costly resource. Automatic post-editors aim at performing the same task, correcting errors produced by the machine translation systems, but at lower cost than humans. Statistical post-editing in particular uses statistical machine translation methods to correct and improve the output of machine translation systems.

The earliest studies on SPE can be traced back to (Allen and Hogan, 2000), who use a parallel corpus composed of three tiers: the source text, its automatic translation and the manually post-edited (i.e. corrected) automatic translation. This study later inspired the original work on SPE by (Simard et al., 2007), who used the Portage System (a PBSMT system) to automatically post-edit the output of an RBMT system, using the raw RBMT output and the manually post-edited (i.e. corrected) output as "source" and "target" side, respectively, of the SPE training data.

SPE systems are generally monolingual, treating the output of a given MT system as the input, and using either manually post-edited or bitext data as the reference translation, as illustrated in Figure 2.1.

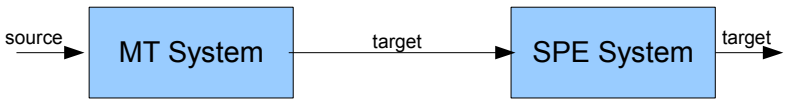

Figure 2.1: Statistical Post-Editing Pipeline

Statistical post-editing has been used to correct the output of rule-based machine translation systems to varying degrees of success, and to a much lesser degree, SPE has been used to post-edit SMT systems. In some approaches, this focuses on

directly negotiating between specific errors in the first stage MT output and the corresponding manual corrections. In other approaches, the second-stage system is instead trained on independent bitext data.

2.3.1 SPE with Manually Post-Edited MT Output

As the main idea behind SPE for MT is to capture the mistakes made by the MT system and to automatically correct them, many methods use first-stage manually corrected MT output to train the second-stage (SPE) system. Several studies have been conducted by combining RBMT (as the first-stage MT system) with PBSMT (as the SPE system), as illustrated in Figure 2.2.

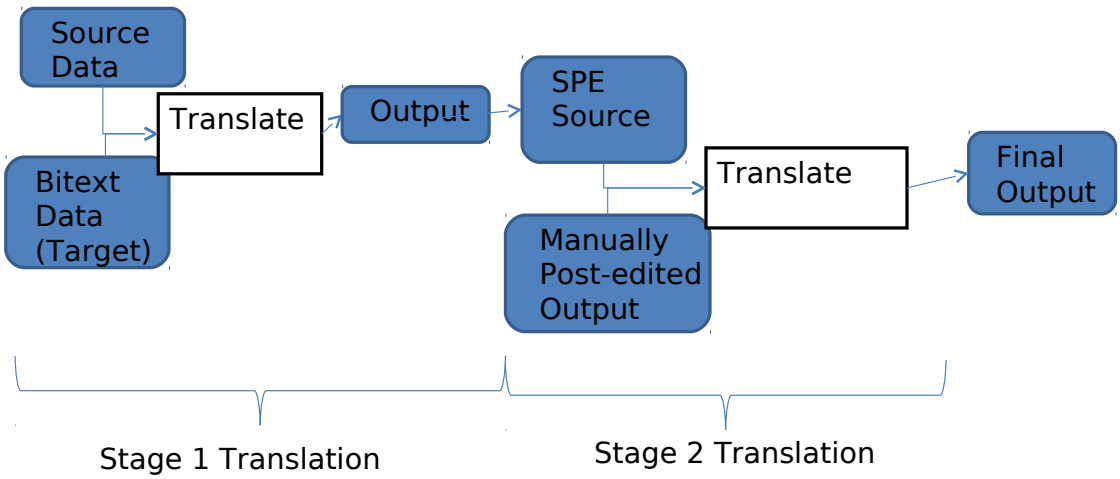

Figure 2.2: Statistical Post-editing using manually post-edited MT out as a reference translation.

The effectiveness of SPE using manually post-edited MT output has been reported in several papers. (Simard et al., 2007) used a phrase based statistical machine translation system (Portage) to post-edit the output of a rule-based machine translation system. In this set-up, manually post-edited RBMT output is used as

the reference translation for the training data of the second-stage "mono-lingual" post-editing system. These experiments, conducted on the Human Resources and Social Development Job Bank, showed that applying the SPE system to RBMT yielded scores higher than each of these two systems on their own. (Simard et al., 2007) also attempted to use the Portage system to post-edit its own output. They reported no gains from using this method. (Isabelle et al., 2007) conducted a set of follow-up experiments in order to adapt RBMT output to a specific domain using Portage as the SPE system. They reported gains of up to 20 BLEU points over the RBMT system alone.

Similar work was done on combining SYSTRAN system with PBSMT systems Moses and Portage (Dugast et al., 2007). Evaluations on the difference between the raw SYSTRAN output and the SYSTRAN+SPE output showed significant improvement in terms of word sense or alternative translation of words, but very low improvement in grammatical and re-ordering categories.

More recently, (Potet et al., 2011) combine a full PBSMT pipeline (SMT+SMT) for translation and post-editing from French to English. The first system translates the French text into English. The MT output is then manually post-edited and introduced into the pipeline following three approaches:

- as supplementary material to enrich the training corpus used to build the translation model,
- as the target side of the parallel corpus used to build the post-editing model,
- as a the target side of the development corpus used to optimize the translation model components weights.

This preliminary study shows a slight improvement over a standalone MT system, but further experiments on larger corpora are needed in order to obtain significant results.

2.3.2 SPE with Independent Bitext Data

An alternative approach uses independently available bitext data (such as Translation Memories), rather than manually corrected first-stage MT output, in SPE pipelines, as shown in Figure 2.3. This method is often less expensive, as the bitext data is already available, and in many cases has also been used by the first-stage system (in scenarios where the first-stage system is also an SMT system). In this group, it is not guaranteed that a divergence between first stage MT output and the target side of the bilingual training data actually corresponds to a translation mistake by the first stage MT system. In several cases, the reference translation may just be a paraphrase of the otherwise good MT output, and not necessarily a correction.

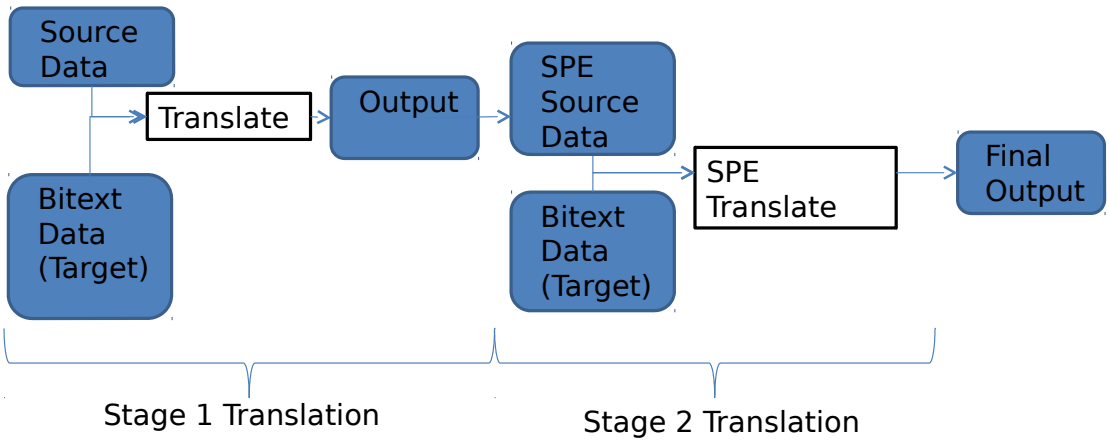

Figure 2.3: Statistical Post-editing using the Translation Memory reference as the reference translation

(Terumasa, 2007) combined RBMT with SPE to translate patent texts, which tend to be difficult to translate without syntactic analysis. Combining SPE with RBMT produced an improved score on the NIST evaluation compared to that of RBMT

alone.

Additional work on the subject was explored by (Lagarda et al., 2009)). As with the previous work, these experiments aimed to improve the output of a commercial RBMT system by using SPE. In this case, the post-editing was carried out using the Moses tool-kit. Experiments were run on two corpora, the Parliament and Protocol corpora, and Moses was trained using the RBMT output as source. The results showed an improvement in the Parliament corpus, the less complex of the two corpora.

(Kuhn et al., 2010) compare the two SPE approaches: the first using manually post-edited MT output and the second using the target side of the bilingual training data. They use Systran RBMT and Portage PBSMT systems, and combine them into a post-editing pipeline, with the RBMT system as first stage and the PBSMT system as the SPE system. The SPE system shows a gain of 10.2 Bleu points compared to the RBMT system alone, on a French-to-English translation task. However, the authors also show that a PBSMT system alone can reach results similar to those obtained by the post-editing pipeline.

(Potet et al., 2012) further investigated these two approaches and showed that SPE systems trained on bilingual training data could not improve over the baseline PBSMT system. However, systems trained on manually post-edited system outputs showed a small improvement in translation quality.

The use of an SMT system to post-edit its own output is not as well explored, however. (Oflazer and El-Khalout, 2007) use statistical post-editing in their experiments exploring selective segmentation based models for English to Turkish translation. They refer to this post-editing as model iteration. They train a post-editing SMT model on the training set decoded by the first stage SMT model and iterate the approach, post-editing the output of the post-editing system. Bleu results show positive improvements, with a cumulative 0.46 increase after 2 iterations. However, there is no indication that these improvements are statistically significant. In (Rubino et al., 2012), a statistical post-editing system is

used to adapt out-of-domain machine translation systems to a specific domain. Their results show that a generic MT system can be adapted through an automatic post-editing step.

2.4 Machine Learning for Quality Estimation

In order to assess the quality of machine translation output, developers rely on a variety of techniques ranging from costly yet accurate, to efficient and easy. While human evaluation is still considered the best and most reliable judgement in quality estimation, this method is inefficient, especially when large corpora are involved. Automatic quality estimation tools have been developed to estimate MT output quality. Automatic evaluation metrics can be divided into sub-groups: those that use reference translations by human evaluators to produce an evaluation, and those that rely solely on the source and hypothesis translation to estimate translation quality.

2.4.1 Translation Evaluation Metrics

Translation evaluation metrics compare machine translation output to a number of reference translations, and return a score that supposedly mimics human judgement.

One of the most popular evaluation metrics in MT development is BLEUPapineni et al. (2002). BLEU matches n-grams between the MT output and the reference translation, using n-gram precision with a brevity penalty as the score, as demonstrated in (2.10)

$$\text{BLEU}(n) = \prod_{1}^{n} PREC_i{}^{\frac{1}{n}} \cdot bp \tag{2.10}$$

where n is the order of n-gram, $PREC_i$ is the i-gram precision and bp is the

brevity penalty. The brevity penalty is defined as (2.11):

$$bp = exp(max(\frac{len(Ref)}{len(Out)} - 1, 0)) \tag{2.11}$$

where $len(Ref)$ is the length of the reference and $len(Out)$ is the length of the output.

This n-gram matching scheme makes BLEU very sensitive to small changes in the output, and fails to capture linguistic variations, especially in the case where only one reference translation is being used. (Callison-Burch et al., 2008) show that that BLEU has a lower correlation with human judgement than metrics such as METEOR, which take into account linguistic resources and better matching strategies. Furthermore, BLEU is designed to evaluate MT output on a document level, and does not fare as well when evaluating quality at a sentence level. TERSnover et al. (2006) is an Edit Distance-style evaluation metric that measures the amount of editing that a human post-editor would have to perform to change a system output so it matches the given reference translation. It calculates how many insertions, deletions, substitutions and sequence shifts are required to make the output identical to the reference. TER is defined in equation(2.12):

$$\text{TER} = \frac{\#INS + \#DEL + \#MOD + \#SHIFT}{len(Ref)} \tag{2.12}$$

2.4.2 Translation Confidence Estimation

Automatic evaluation metrics require a sentence-for-sentence reference translation in order to assess the quality of machine translation output. While these metrics reflect human judgement quite well, they usually do so at the corpus level, and have difficulty assessing the quality of a single sentence. Furthermore, they are restricted by the need for reference translations. Confidence estimation assesses the translation quality of a given machine translation's expected output quality by analysing the source text, the hypothesis, and other relevant information, but

without access to a reference. With enough training, successful quality estimation methods can evaluate MT output without having access to a reference translation. By doing so at a sentence level, these machine learning techniques can be used to determine the quality of a given translation output, learn from comparing it to its post-edited counterpart, and decide whether or not unseen translations benefit from post-editing. Quality Estimation techniques can also be used in system combination, where a collection of different systems is available. Confidence estimation probabilities can provide a convenient way of combining the outputs of different systems.

The use of quality estimation is widely used in speech recognition, but is increasingly used in machine translation, and its methods are employed in sentence selection, predicting post-editing effort, and even to improve components of existing systems.

Confidence estimation can be further divided into two steps, feature selection and machine learning.

Predictor Features

Confidence estimation generally relies on a feature vector which encapsulates information about the hypothesis text, and a variable that indicates the quality of the translation. This variable can range from a binary value that indicates whether the translation is good or bad, or a continuous score that assesses the translation quality. The features in question depend on the source and hypothesis text, but not the reference itself. This enables testers to use confidence estimation in situations such as ours, where the reference translation is unavailable at testing time.

The predictor features used in confidence estimation systems can be MT system dependent or system independent features. System independent features treat the machine translation system as a black box and instead look at surface properties of the source text and the hypotheses text. These features depend heavily on the task in question, and can vary from simple linguistic features such as n-gram

distributions, to language model scores and perplexity of sentences. More in-depth system independent features may encapsulate syntactic information such as part of speech data.

System dependent features are features that depend on the machine translation system, and rely on having access to the translation process itself, and not just the input and output of the system. Relying on system-dependent features might be problematic in the use of commercial systems, where the inner workings of the machine translation process are not available for study. In open-source statistical machine translation systems, for example, these features might include phrase-translation model scores, distance and lexical based reordering model scores, and the word penalty.

In (Specia et al., 2009), the problem of predictor features is investigated at great lengths in order to identify the most consistently relevant features. These 17 baseline features are system-independent, and are generally geared towards prediction of post-editing effort, but have proven useful in any confidence estimation task. As such, these are the features we will use as our baseline in our confidence estimation and classification experiments.

Machine Learning

Several different machine learning algorithms have been proposed and employed in order to adequately combine predictor features. Given a feature vector and a label x, these machine learning algorithms return a probability estimate of correctness. These algorithms include naive Bayes, Bayesian nets, neural networks, boosting, linear models, decision trees, and support vector machines.

(Sanchis et al., 2003) use a smoothed naive Bayes classification model to combine predictor features in order to classify speech utterances as either correct or incorrect. Later, (Blatz et al., 2004) applies the same model to confidence estimation for machine translation output both at sentence and sub-sentence level.

(Guillevic et al., 2002) use a Multi-Layer Perceptron (MLP) classifier trained to

discriminate between correct and incorrect concepts for dialogue systems. (Moreno et al., 2001) employ the use of a a boosting classification algorithm for confidence scoring in a speech recognition task. In the field of machine translation, the first descriptions of machine learning methods are in (Ueffing et al., 2003), which include posterior probabilities for estimating the machine translation correctness on a word level. (Gandrabur and Foster, 2003) investigate the use of machine learning algorithms in machine translation, and describe the use of a neural-net confidence estimation layer in an interactive translator's tool.

(Ma et al., 2002) train and test a Support Vector Machine (SVM) classifier and compare the results with other statistical classification methods.

(Blatz et al., 2004) introduced the use of quality estimation on the sentence level by using MT metrics to determine the "correctness" of the machine translated output. Using Word Error Rate and NIST scores, they label sentences as "good" or "bad", and estimate the quality of the output sentences by analysing features in the source and target texts. In addition to the classification of sentences into good or bad, a regression model is trained that attempts to estimate the scores.

(Specia et al., 2010) extend quality estimation beyond predicting a binary score of "good" and "bad", and use it to predict a score in a given range. Their research makes use of classification and regression algorithms and system independent features in order to predict the quality of translations at a sentence level.

(He et al., 2010a) use quality estimation to predict human post-editing effort and recommend the SMT outputs to a Translation Memory user based on estimated post-editing effort. Our own work on quality estimation builds off the research presented in this section, especially the work of (He et al., 2010a), which focuses on post-editing effort.

Chapter 3

Statistical Post-Editing

3.1 Introduction

Statistical post-editing (SPE) has been shown to successfully improve the output of Rule-Based MT (RBMT) systems, and RBMT-SPE pipelines are already a commercial reality. The impact of SPE on Statistical Machine Translation (SMT) output, is less well-researched, however.

This section describes a set of experiments in which we follow the overall design put forth by (Dugast et al., 2007), where an SPE system is trained on the output of the first-stage RBMT system. This SPE system is also applied to the output of an SMT system. Our experiments always use pre-existing bitext data rather manually post-edited MT output to train our second-stage SPE system. The objective is to extend the previous work on SPE and investigate whether or not SPE can improve on SMT using the same SMT engine. We also introduce a novel context-aware approach to post-editing, which preserves some source context information in the post-editing step. This approach shows improvements for the Translation Memory described in Section 3.2.1. During the early parts of our research, published in Béchara et al. (2011), we used Translation Memory based data from one of the CNGL Industry partners as our data sets. However, at a later stage of our research, when we attempted to extend our experiments to a publicly

available corpus, in order to reproduce our earlier results, we were unable to achieve the same improvements to our test set. This caused us to reinvestigate the data splits (training, development and test) in the earlier experiments using the Translation Memory based data sets. This investigation revealed a bias in the test data set. We were not able to duplicate the positive results on a randomly extracted test set from the same Translation Memory data.

3.2 Data and Tools

3.2.1 Symantec Translation Memory

In this thesis, for the most part, we have used the same data throughout our experiments, in order to be able to compare our results. This data is part of an English-French translation memory provided by Symantec. The data is part of a very tight domain (technical software user help information), which is reflected by its relatively small vocabulary size. After removing all TMC markup and meta-information, in order to make the data ready for our translation software, we extracted 53,000 unique sentences. From this data we later randomly extracted 50,000 French-English sentence pairs to be our training set. The sentences are between 1 and 98 words in length for English, and 1 and 100 words in length for French. The average sentence length in the training set is 13 words for English and 15 words for French, with a vocabulary size of 9,273 for the English side of the data, and 12,070 for French. The remaining sentences were split into a test set of 1967 sentences, and a development set of 972 sentences.

3.2.2 Statistical Phrase-Based Machine Translation

For our statistical machine translation system we used the PB-SMT system Moses, (Koehn et al., 2007), 5-gram language models with Kneser-Ney smoothing trained with SRILM, (Stolcke, 2002), the GIZA++ implementation of IBM word

alignment model 4 (Och and Ney, 2003), with refinement and phrase-extraction heuristics as described in (Koehn et al., 2003). We used minimum error rate training (MERT) (Och, 2003) for tuning on the development set. During decoding, the stack size was limited to 500 hypotheses.

3.2.3 Systran Machine Translation System

As our rule-based machine translation system for the first stage MT in our experiments, we used the Systran Enterprise Server 6 production system, specifically customised with the use of 10K+ dictionary entries specific to the text type and domain of the Symantec translation memory data, as described in (Roturier, 2009).

3.3 Statistical Post-Editing for a Rule-Based Machine Translation System

Our experiments follow the design put forth by (Dugast et al., 2007), where the output of the RBMT system is used to train a monolingual second-stage (SPE) system. In this case, the RBMT system is used to translate the entire training set, and this output is used as a source-side training set for the second-stage system, as illustrated in Figure 3.1. Here, the target side of the training set (f) is derived from the Translation Memory. The second-stage SMT system therefore builds a translation model using the output of the RBMT system as a source, and the Translation Memory's reference translation as the reference.
Using Systran as our RBMT system and Moses as our SMT system, we implemented the pipeline and tested it on the held out Translation Memory data provided by Symantec. The results are detailed in Section 3.3.1.

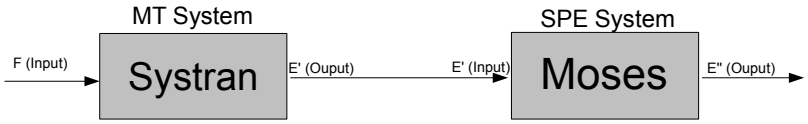

Figure 3.1: The RBMT+SPE pipeline, using the output of RBMT as the input for the second stage SMT system

	RBMT	SMT	RBMT+SPE
BLEU	23.26	65.43	64.63
TER	61.07	23.92	24.62

Table 3.1: BLEU and TER scores for the RBMT, SMT and the SPE systems (French to English)

3.3.1 Experimental Results

Experiments run on the 1967 sentence test set of the Symantec Translation Memory show that applying SPE to RBMT can improve the output in terms of Bleu and Ter evaluation scores. Table 3.1 shows spectacular Bleu score improvements over the RBMT system on its own. The RBMT system on its own scores a dismal 23.26 Bleu points. But after applying the SPE system, the combined RBMT+SPE system scores rival that of the SMT system on its own. However, the results suggest that even though SPE improves over RBMT, it still does not outperform the SMT system. Because automatic metrics directly compare our output to a reference translation, they tend to favour the output of statistical machine translation systems. This also explains the low RBMT score, despite reports that human evaluators rate the Systran Machine Translation system as quite appropriate.

3.4 Statistical Post-Editing for a Statistical Machine Translation System

Section 3.3 described the RBMT+SPE pipeline using the output of the RBMT system and the bitext data reference translation to train the second-stage system. In our second pipeline, the same SMT system is used throughout both stages, as demonstrated in Figure 3.2.

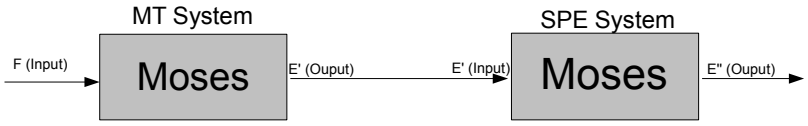

Figure 3.2: The SMT+SPE pipeline, using the output of Moses as the input for the second stage SMT system (Moses)

The training of the second-stage system is more problematic, however, as we cannot simply use the SMT system to translate the its own training data to generate the monolingual source-side for the SPE system. In order to obtain this new "source" training data for the second-stage mono-lingual PB-SMT system, we need to train systems using a 10-fold cross-validation approach on the training set. By translating each 10% of the data based on a system trained on the remaining 90%, we avoid translation of the already seen data in the creation of the second-stage system's training data. Figure 3.3 demonstrates the 10-fold cross-validation method used to create the new source-side training set, otherwise known as f'. The mono-lingual system is trained using f' as a source side training data, and the bitext reference translation from the TM (the same one used in the first-stage system), as a reference translation. The same test and development sets are used throughout for decoding and tuning respectively, and are individually translated in each step, using the full first-stage SMT and second-stage SPE training sets.

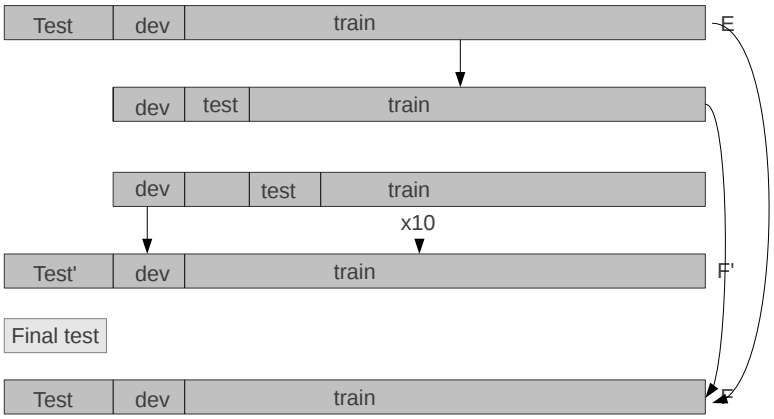

Figure 3.3: The ten-fold cross-validation model used to create the source-side training set for the SPE system

3.4.1 Contextual SPE

In our basic SPE pipeline (PE), the second-stage SPE system is trained on the output (f') of the (10-fold cross validation version of the) first-stage MT system, effectively resulting in a "mono-lingual" SPE system (f'-f). In a sense, however, the second-stage SPE system has lost the connection to the original source data: ideally we would like to be able to be in a position to distinguish between situations where f' is a good translation of some source word (or phrase) e, and situations where f' should be post-edited to f. In some of the experiments reported, we model this by recording the source word (or phrase) e that gave rise to f' as f'#e (i.e. concatenating f' with # and e), effectively creating a new intermediate language F'#E as the source language for a context-aware second-stage SPE system (PE-C). In our experiments we do this using GIZA++ word-alignments as illustrated in the following example:

- Source E: `if an original file has been deleted , but backup files are still available ...`
- Target F: `si un fichier original a été supprimé , mais si les fichiers`

`de sauvegarde sont toujours disponibles ...`

- Baseline Output F': `si un fichier initial a été supprimé , mais les fichiers de sauvegarde sont encore disponibles ...`
- Context F'#E: `si#if un#an fichier#file initial#original a#has été#been supprimé#deleted ,#, mais#but les#files fichiers#files de#backup sauvegarde#backup sont#are encore#still disponibles#available ...`
- PE-C Output F": `si un fichier original a été supprimé , mais les fichiers de sauvegarde sont toujours disponibles ...`

Here, the baseline output `initial` and `encores` was changed to `original` and `toujours`, ensuring a better match with the target text.

Thresholding Context Information by Alignment Strength While this new intermediate language preserves context information, the vocabulary size increases from 9273 in the EN training set to 70780 in the F'#E training set. This increase, and the ensuing data sparseness, have potentially adverse effects on translation quality. Furthermore, the word alignment data used to create this new language is not always reliable. In order to address the issue of data sparseness and unreliable word alignment data, we carried out experiments restricting the amount of context information available to PE-C systems. In particular, we used GIZA++ word alignment strengths to filter context information, using the word alignment levels of $\geq$0.6, $\geq$0.7, $\geq$0.8 and $\geq$0.9 as thresholds: that is for each threshold, source words that are aligned with translation output words with an alignment score greater than or equal to the threshold are used as source context words e in f'#e pairs for the source side of the second stage PBSMT SPE system.

3.5 Experimental Results

In this section we present results for English to French and French to English translation and post-editing experiments.

3.5.1 English to French

Results Using SPE

In order to evaluate our SPE approach, we train two PBSMT systems for a post-editing pipeline, the first stage system (Baseline) between E and F, producing output F' given input E, and the second stage mono-lingual post-editing system between F' and F, producing output F" given F' as input. We train post-editing pipeline systems without (PE) and with (PE-C) context information.
Table 3.2 shows that simple PE fails to improve over the Baseline and that the drop in the BLEU score for the PE-C (post-editing with context information) compared to the Baseline (and PE) is marked. The most likely reason for this drop is the explosion in the size of the vocabulary set between E and F'#E in the post-editing with context information setting (PE-C). This is visible in the output of the second stage post-editing system in the form of untranslated f'#e items. These are effectively OOV (out-of-vocabulary) items that the second stage system has not encountered during training. As the f' part of an f'#e item is already a word in the target language, we simply filter the f'#e items in the output by automatically deleting the source context information suffix #e from such items. This is illustrated in the example below:

- PE-C: `dell recommande de renseigne#populate la baie de disque avec les disques physiques de la même capacité .`
- PE-CF(filtered): `dell recommande de renseigne la baie de disque avec les disques physiques de la même capacité .`

We refer to this output as PE-CF. The BLEU score for PE-CF is much closer to the Baseline than that for PE-C. In all our experiments reported in the remainder of this paper we use this simple output filtering prior to evaluating Context-Informed SPE models.
Overall results show that for our data set, a simple second-stage PB-SMT system (with and without context information) is unable to improve on the first-stage

Score	Baseline	PE	PE-C	PE-CF
BLEU	60.30	60.15	46.89	58.55

Table 3.2: English–French SPE results

PB-SMT system in a pure PB-SMT post-editing pipeline for English to French machine translation.

Thresholding Context Information by Alignment Strength Table 3.3 shows the results for the context aware post-editing pipeline PE-CF with alignment strength thresholding on the full test set.

Threshold	0.6	0.7	0.8	0.9
PE-CF	59.80	60.30	60.23	59.73

Table 3.3: English–French Translation using Contextual SPE with Alignment Thresholding (BLEU scores)

Thresholding shows clear improvements over simple PE-CF in Table 3.2, however, none of them show improvements over the baseline in Table 3.2. Clearly, for the English to French translation direction and our data set, all our PB-SMT SPE pipelines (even those that are context aware and use thresholding) fail to improve on the PB-SMT Baseline.

3.5.2 French to English

We ran the same set of experiments for the other translation direction, French to English.

Experimental Results Using SPE

Simple PE results (Table 3.4) on the test set show that for our data set a simple second-stage PB-SMT system is able to improve on the first-stage PB-SMT system

in a pure PB-SMT post-editing pipeline, with a small increase in BLEU of 0.65 absolute over the baseline. This result is statistically significant[1]. Compared to Baseline and PE, BLEU scores deteriorate for the context-aware post-editing pipeline PE-C, as any beneficial impact of the post-editing pipeline is swamped by data sparseness and OOV items in the output of the second stage PE-C system. The most likely reason for this drop is again the explosion in the size of the vocabulary set between E and E'#F in the post-editing with context information setting: the training set vocabulary size is 9,273 for E compared to 47,730 for E'#F, resulting in both data-sparseness and OOV occurrences for the second stage PB-SMT system in the context informed post-editing pipeline PE-C. Filtering out the #f tags in the output, leaving only the target word, brings the BLEU score up to 61.36 for PE-CF.

Score	Baseline	PE	PE-C	PE-CF
BLEU	61.60	62.25	57.33	61.36

Table 3.4: French–English SPE results

Thresholding Context Information by Alignment Strength Mirroring the English to French Experiments, we carry out experiments restricting the amount of context information available to PE-C systems, filtering context information by thresholding word alignment strengths, using the GIZA++ based word alignment levels of $\geq$0.6, $\geq$0.7, $\geq$0.8 and $\geq$0.9 as thresholds. Results are presented in Table 3.5.

Threshold	0.6	0.7	0.8	0.9
PE-CF	63.76	63.54	63.89	63.80

Table 3.5: French–English Translation using Contextual SPE with Alignment Thresholding (BLEU scores)

[1]Using approximate randomisation methods as implemented in FastMTEval

The results in Table 3.4, when compared to the baseline (61.60) in Table 3.3, show that for all alignment thresholds, for this data set and the French to English translation direction, context aware post-editing pipeline PE-CF results outperform the Baseline by about 2 BLEU points absolute. All results are statistically significant at the $p \leq 0.05$ level.

3.6 Additional Experiments

3.6.1 The JRC-ACQUIS Corpus

We decided to extend our experiments to a more freely available corpus so that our experiments can be replicated with publicly available data, but also in order to find out if the improvements in the SYMANTEC corpus can be reproduced with different data. For this purpose, we used the ACQUIS corpus, a collective of legislative texts which is constantly changing and improving (Steinberger et al., 2006). We re-ran the basic experiments using a PBSMT pipeline. This system was trained on 360,000 unique sentences from the ACQUIS corpus, and then tuned (using MERT) on 1000 different sentences.

3.6.2 Results with Context-Thresholding

The results in Tables 3.6 and 3.7 show the results in the French to English direction on the 2000 unique sentences randomly extracted from the JRC-ACQUIS corpus.

Score	Baseline	PE	PECF
BLEU	67.93	65.29	66.66

Table 3.6: French–English SPE results for JRC-Acquis

The results in Table 3.7 suggest that the context-aware approach does not work as well for the JRC-Acquis corpus as it does for our Symantec Translation Memory data. This might be a result of the less constrained domain of JRC-Acquis,

Threshold	0.6	0.7	0.8	0.9
PE-CF	66.67	66.60	65.39	66.45

Table 3.7: French–English Translation using Contextual SPE with Alignment Thresholding (BLEU scores) for the JRC-Acquis corpus

making the alignment information less useful in the post-editing phase. These results prompted further investigation into the results presented in Section 3.5.

3.6.3 Re-evaluating the Translation Memory

Our previous SPE results show that using a context-aware approach in an PBSMT pipeline produces a statistically significant improvement of 2 BLEU points over the baseline in the direction of French to English. However, further experiments showed that these results did not extend to other language pairs (En-Fr showed little to no change), and other data sets (Section 3.6.1). The improvement observed seems to be unique to a particular data set in a particular language direction. This required further evaluation and investigation in order to determine whether or not this data is replicable and valid.

In order to approach the issue at stake, we chose a new test set from the Symantec Translation Memory and set out to repeat our previous experiments. The new 2000 sentence test set and 1000 sentence development set were selected at random out of the 53,000 sentences that made up the Translation Memory with the remaining 50,000 sentences for training. We repeated the process described in Section 3.4, including the context thresholding and filtering, in order to see if we can replicate the initial results.

Threshold	Baseline	SPE	0.6	0.7	0.8	0.9
PE-CF	62.52	62.61	59.18	61.54	59.56	59.98

Table 3.8: English–French SPE results for new test set (2000)

Threshold	Baseline	SPE	0.6	0.7	0.8	0.9
PE-CF	65.54	65.54	64.39	64.91	64.66	65.29

Table 3.9: French–English SPE results for new test set (2000)

The results show that we are unable reproduce the results presented in Section 3.5, where a combination of alignment information and probability thresholding with statistical post-editing seemed to improve our output score by 2 Bleu points in the French–English direction. The results in Table 3.9 show that SPE with context thresholding actually performs worse than a basic source context-free SPE system, and a pure SMT system. While the resulting drop in quality is not statistically significant, it still is enough to invalidate the results reported in previous experiments.

A further investigation into the original test set itself showed that the sentences used in Section 3.5 were, on average, shorter than those in the new 2000 sentence test set, and contained a significant number of shorter and single-word sentences. This leads us to believe that the first test set was not an adequate representation of the Translation Memory.

Further experiments described in this thesis as using the Translation Memory use the new test and training set randomly extracted in this section.

3.6.4 Monotone Post-editing

Tables (3.10) and (3.11) show the results for a word-based SPE system compared to the Phrase-Based SPE system. The Monotone system was created by simply switching off reordering in Moses, and restricting the Moses phrase-table to 1. In this section, we use the new test set selected in Section 3.6.3.

While the word-based statistical post-editing system seems to show an improvement over Phrase-Based SPE, it still does not show an improvement over the baseline SMT system. A closer look at the sentences that are improving shows

Score	Baseline	PE	PECF
BLEU	67.93	65.79	66.84

Table 3.10: French–English SPE results (Bleu scores) for Monotone SPE

Threshold	0.6	0.7	0.8	0.9
PE-CF	67.05	67.9	67.88	67.89

Table 3.11: French–English Translation using Contextual SPE with Alignment Thresholding (Bleu scores) using Monotone-Based SPE

that only a very small number of sentences (see Table 3.12) are being affected by the word-based SPE system. Table 3.12 shows the number of sentences in the post-editing phase that changed compared to the baseline sentences. Sentences determined to be exact matches count as "same as baseline", while even a minor difference such as punctuation counts as "different to baseline". These sentences show only lexical changes, as expected from a word-based SPE system.

Threshold	0.6	0.7	0.8	0.9
changes	73	77	73	69

Table 3.12: Number of sentences that change with word-based post-editing (out of 2000)

A closer look at the SPE phrase table during the word-based training phase shows that in the case of monolingual translation with no context information, there is no entry where a word is not mapped to itself, except when it is mapped to nothing, which accounts for the drop in quality between the baseline output and the simple post-edited output. In the case of full context information (with no thresholding), however, there is no instance where the opposite is true. In the case of context thresholding, the majority of the entries are different (89%), and only a small minority are the same (0.11). These results are expected considering the

context information, which creates the new language. If we were to remove the glue in the phrase table, our results would show a monolingual translation, with most of the phrase table (or in fact all of it) pointing to itself.
The context-thresholding word-based statistical post-editing system seems to be able to improve the output lexically, but the improvement is so minimal, and in so few sentences, that it is not significant enough to make a difference over 2000 sentences.

3.7 Summary

The use of SPE to improve the output of RBMT systems is already being used commercially in certain Systran releases. The impact of SPE on statistical models has been less thoroughly researched. In this chapter, we investigated the possibilities of applying SPE to SMT output and implemented a novel context-modelling approach, which uses GIZA++ word alignment information to preserve context information in the second-stage mono-lingual translation phase. We followed the experimental design of (Kuhn et al., 2010), using bitext data to post-edit the output of an RBMT system (Systran). Our preliminary experiments appear to confirm what (Kuhn et al., 2010) reported in their findings. Using SPE on the output of an RBMT system can dramatically improve the quality of the translation, at least in terms of automatic evalation metrics such as Bleu and Ter. However, automatic evaluation metrics seem to show that SPE does not improve RBMT systems over the pure SMT system (Table 3.1). Additionally, we notice that the automatic evaluation metrics overwhelmingly favour the systems that use statistical models, including both the SMT system and the RBMT+SMT system.
Furthermore, we applied our SPE system to SMT (Moses) output and found that in general, it fails to improve over the baseline to a significant degree. Introducing the concept of context-modelling with alignment thresholding, we managed to

improve the output for a specific test set in a specific language direction (French to English). However, these results could not be replicated with publically available JRC-Acquis data. Further investigation showed that the original 1967 segment Symantec TM test data were not fully representative of the overall TM data and earlier improvements could not be when a more random test set was selected from the Translation Memory.

Chapter 4

Human Evaluation and Error Analysis

4.1 Introduction

Despite the speed and efficiency of automatic evaluation metrics such as BLEU and TER, human evaluation is still considered the gold standard in evaluation of translation systems. Automatic evaluation metrics are still considered to be an imperfect substitute, and shared tasks such as the annual Workshop on Machine Translation (WMT) still define human evaluation as primary and use their judgements to evaluate automatic metrics.

In Chapter 3 we showed that SPE can dramatically improve the output of the RBMT Systran in terms of Bleu and Ter scores. These scores, however, compare the output to a reference translation. This reference translation is a part of the Translation Memory, a subset of which is also used to train our SMT model. In many instances, therefore, the SPE system we are using not only corrects, but also paraphrases our MT output. This pushes the output closer to the reference translation, not necessarily producing a better output, but causing improved Bleu and Ter scores than the RBMT system on its own, which itself is not trained on any parallel corpus.

The example below shows how the SPE system can often paraphrase, without really improving the RBMT output.

- *Source*: si cela ne corrige pas le probléme , *vous pouvez devoir* appeler le support technique de symantec pour corriger les entrées de registre du produit

- *Reference*: if this does not correct the *issue* , you may need to call symantec technical support for assistance in correcting *the product 's registry entries* .

- *RBMT*: if that does not correct the *problem* , you can have to call the customer support of symantec for an assistance to *correct the registry entries of the product* .

- *SPE*: if it does not correct the *issue* , you may need to call the customer support of symantec for an assistance to correct *the product's registry entries* .

The RBMT translation *the registry entries of the product* is valid. However, the SPE system still post-edits it to read *the product's registry entries*. Since this phrase actually matches up with the reference translation, the SPE sentence will result in a higher score using our automatic evaluation metrics, despite the fact that is not a correction, but simply a paraphrase. Also of interest is the word *probléme*, which the RBMT system translates as *problem*. The SPE "corrects" this word to *issue*, an equally valid choice in this context. But while *problem* is not an incorrect choice, the automatic evaluation metrics will still consider *issue* to be more valid, because it matches up with the reference translation.
The above example demonstrates some of many cases where the automatic evaluation metrics are biased in favour of the statistical output, whether it is from a pure SMT system or from the post-edited RBMT output. In this chapter, we turn to human evaluators in order to investigate the true quality of the post-editor without the bias of the reference translation.

4.2 The Evaluation Environment

In order to gain some insight into the actual impact of SPE on the RBMT system, and perhaps confirm the seemingly spectacular improvements that the automatic evaluation metrics seem to suggest, we enlist the help of human evaluators.
We selected, at random, 200 unique sentences from our Symantec TM data and translated them by each of our four systems: pure SMT, pure RBMT and the two SPE pipelines RBMT+SPE and SMT+SPE. Sentences that were translated identically were not shown to the evaluators, but instead automatically marked as "equal quality". We designed an evaluation environment to present these sentences and their translations to our evaluators. The environment is a web application developed using Python tools and the Django framework, and is available at http://speval.yifanhe.org. Each evaluator was given a username and a password to log into the system. Once they have logged in, they are shown a source (French) sentence and two output (English) sentences from different systems, the latter two presented in a random order. Once the evaluator chooses the better sentence, or a tie in case of equal translation quality, they proceed to the next page, which presents a new sentence. A screenshot of the task is available in Figure 4.1. The tool records each choice, changes in choices, and a time stamp for the time at which the evaluator made their choice for each sentence. This helps give us additional data on the difficulty of the task, and the confidence of the evaluator in each of their choices.
Evaluation was carried out by ten different translators of varied backgrounds. All of these translators are bilingual and fluent in both French and English. Six out of the ten evaluators are native speakers of French, and the others have a good grasp of French, evidenced by school and professional certificates. While none of them are professional translators, all of them have experience with machine translation or localisation, or are masters students in translation. All the evaluators were fully briefed on the task and were given a chance to conduct a test run.

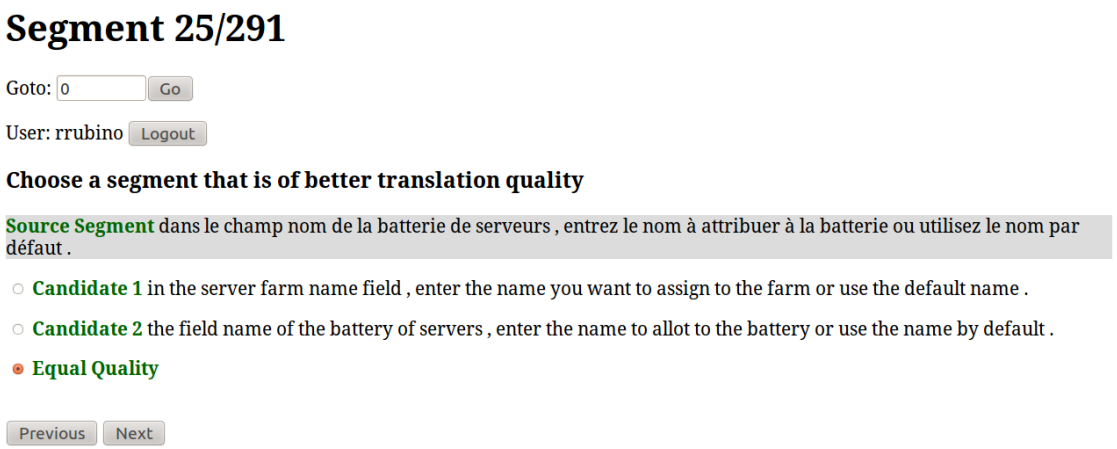

Figure 4.1: A screen-shot of the manual evaluation task at http://speval.yifanhe.org, using python tools provided by Yifan He.

As demonstrated in Figure 4.1, the evaluators were shown a source sentence (in French) and asked which of the two MT outputs (presented in random order) is a better translation, or if they are of equal quality. In order to avoid biasing the evaluator, we did not provide a reference translation at any point during the evaluation task, and instead relied on the evaluators' judgement of each sentence's fluency and adequacy. The task was available to be completed online, and evaluators could save their progress and return to the task at any time. The subjects were paid for their time and were given a week to submit the task, which did not have to be completed in one sitting. The evaluators generally rated the task as difficult, especially as the domain was highly technical and the sentences often fragmented and containing a large number of symbols and abbreviations.

4.2.1 Annotator Agreement

For these results to be meaningful, a reasonable degree of agreement must exist between evaluators to support the validity of our human evaluation experiment. In order to measure this agreement, we calculated pair-wise inter-annotator agreement between all of the different evaluators.

For this agreement, we used Cohen's κ measure. κ is a more robust measure

compared to simple percent agreement calculation, as it takes into account the agreement occurring by chance. κ is defined by the formula in 4.1.

$$\kappa = \frac{Pr(a) - Pr(e)}{1 - Pr(e)} \tag{4.1}$$

Pr(a) is the proportion of times two annotators are observed to agree, and *Pr(e)* is the expected proportion of times the two annotators are expected to agree by chance. Agreement occurs when two annotators compare the same systems and agree on their rankings. In our case there are three possible choices; either one system is better than the other, or it is worse, or there is a tie. κ ranges between 0 and 1, with 1 indicating a higher rate of agreement, and 0 indicating low or no agreement.

According to (Landis and Koch, 1977), a moderate agreement falls between 0.4 and 0.6. A substantial agreement falls between 0.6 and 0.8, and 0.2 to 0.4 indicate a fair agreement, while anything below that is considered slight. Full results for all ten evaluators ($\kappa = 0.42$) are on the border between moderate to fair. As two of our evaluators scored an average agreement under 0.4, we discarded their results as weak and used only the results from the 8 evaluators that had a moderate agreement. Without the outliers, our average agreement for the evaluators is 0.47. This amounts to a moderate agreement.

4.2.2 Human Evaluation Results

The system kept score of each time a sentence was chosen as the "best translation" by one of the 8 remaining evaluators. These results, once added together, were normalised based on the number of evaluators. We compared these results to the number of times the sentences were chosen as the "best translation" by the automatic evaluation metrics. We compared the S-Bleu and Ter scores for the sentences of each of the outputs, and tallied up the number of times each system was given the best score by either S-Bleu or Ter. We used S-Bleu(Lin and

Och, 2004) instead of BLEU in order to evaluate the document at a sentence level. S-BLEU will still positively score segments that do not have higher n-gram (n=4 in our setting) matching, unless there is no unigram match.

We then compared the results from the human evaluators with the S-BLEU and TER results.

When comparing RBMT with SMT (Figure 4.2), the automatic evaluation metrics very rarely chose the RBMT system as the best system. S-BLEU only chooses 16 out of 200 sentences from the RBMT output, and TER only chooses 9. Human evaluators, on the other hand, choose the best sentence from the RBMT output 50 times. This shows that while human evaluators, in this specific case, still seem to prefer the SMT output over the RBMT output, the RBMT system still performs nowhere near as dismally as the automatic evaluation metrics seem to suggest. Human evaluators only prefer SMT output sentences in about half (97) of the 200 sentences, and in the other half either choose RBMT or a tie.

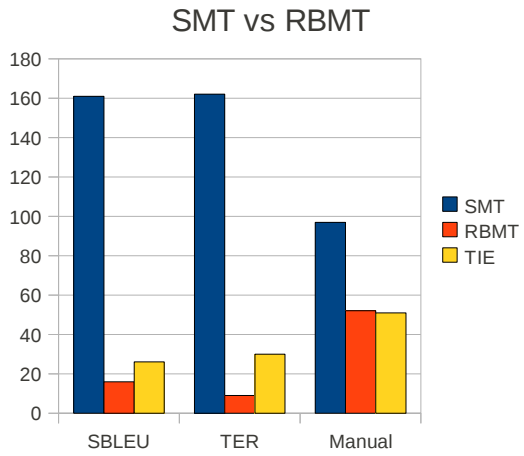

Figure 4.2: SMT vs RBMT comparison using BLEU, TER and manual evaluation

Similar discrepancies between automatic and human judgement can be seen in the

comparison between SMT and RBMT+SPE (Figure 4.3). S-BLEU and TER seem to favour SMT output sentences over the post-edited RBMT output (RBMT+SPE). Human evaluators, however, chose the post-edited output in 40 out of 200 sentences, and chose the pure SMT system only 28 times. In the rest of the sentences (138), human evaluators judged the outputs as ties. This indicates that the RBMT+SPE system can perform at least as well, if not better, than the SMT system on its own, based on human judgement.

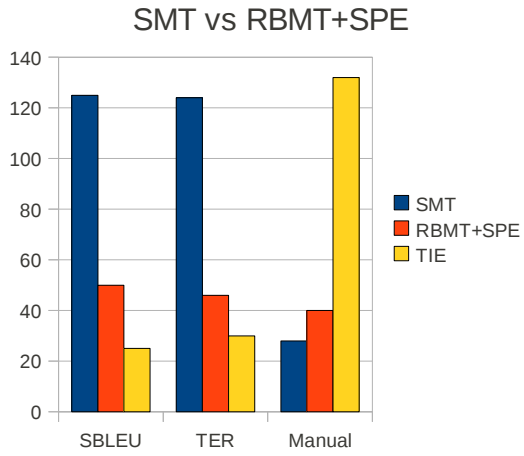

Figure 4.3: SMT vs RBMT+SPE comparison using BLEU, TER and manual evaluation

The comparison between RBMT and RBMT+SPE (Figure 4.4) shows that while human evaluators and automatic metrics are in agreement that post-editing generally improves the system, human evaluators seem to disagree on the extent to which it impacts the results. Automatic metrics seem to overwhelmingly favour the post-edited system, and TER only chooses RBMT sentences in 11 out of 200 cases. Human evaluators, however, choose the RBMT output sentences in 40 of the sentences. This is a reflection of the bias that the automatic metrics show for

the statistical system.

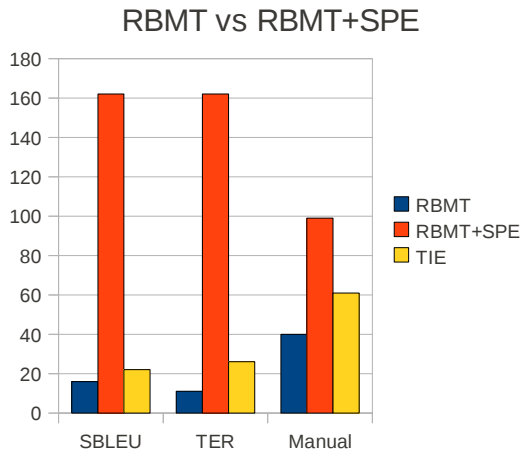

Figure 4.4: RBMT vs RBMT+SPE comparison using BLEU, TER and manual evaluation

4.2.3 Evaluation Time and Task Difficulty

In order to assess the difficulty of comparing different systems, we recorded the time each evaluator spent evaluating a translation pair. We assume that spending more time on an evaluation indicates that it is more difficult to select the best translation. We report averaged results in Figure 4.5. The results show that comparing the two stand-alone MT systems (RBMT and SMT) takes an evaluator, on average, more than 20 seconds to reach a decision. We therefore conclude that this is a difficult task. This is most likely due to the profound differences in terms of syntax and vocabulary between the SMT and the RBMT outputs. By contrast, when comparing SMT versus SMT+SPE, the time spent drops by nearly 10 seconds (on average). This is most likely due to the fact that SMT and SMT+SPE outputs are very similar and therefore require less time to scan and judge. A

similar trend can be observed when comparing SMT with RBMT+SPE, where again outputs are more similar than between SMT and RBMT on average. Finally, choosing between RBMT and RBMT+SPE requires the least amount of time. This is consistent with the observation that (according to the human evaluation) the quality difference between RBMT and RBMT+SPE is the most pronounced, and therefore more "obvious" than in the other cases.

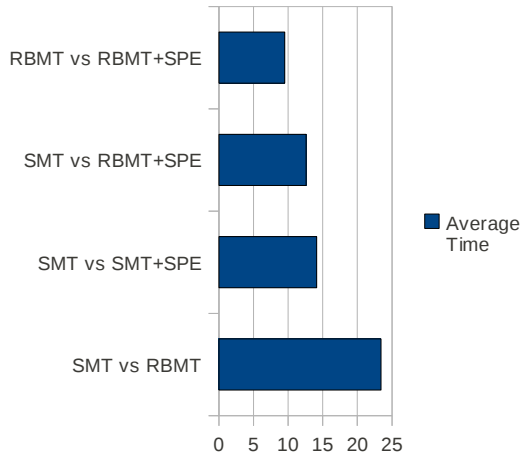

Figure 4.5: Average time spent (in seconds) by human evaluators on each system comparison

4.3 Error Analysis

The results detailed in Section (4.2.2) showed that SPE can improve over a baseline RBMT system when applied to the output. These results also show a discrepancy between the human evaluation results and the automatic metrics. In order to obtain a better understanding of the translation quality gains between the RBMT system and the RBMT+SPE system, and to gain insight into why there are discrepancies between the manual and automatic evaluation results, we

performed an additional manual sentence-level error analysis in a bid to reveal the advantages and disadvantages of the SPE pipelines compared with the RBMT and SMT systems.

4.3.1 Automatic Error Analysis

As TER is an edit distance based evaluation metric, a closer look at its scores might give us a little more insight into the nature of the changes between the RBMT system and the post-editing system. TER edit statistics are divided into four categories:

- Insertion (Ins): instances where words that do not exist in the reference translation have been inserted into the hypothesis translation
- Substitution (Sub): instances where words that exist in the reference translation are substituted for different words in the hypothesis translation
- Deletion (Del): instances where words that exist in the reference translation are not present in the hypothesis translation
- Shift: instances where the word is present in both the reference and hypothesis translation, but not in the exact same place in the sentence

We extracted these edit statistics for each the systems we are comparing: SMT, RBMT, SMT+SPE and RBMT+SPE outputs. We summarise our results in Table 4.1. The numbers have been normalised using sentence length to make them comparable.

Table 4.1 suggests that applying SPE to the RBMT system achieves significant gains in the insertion and substitution categories, and to a lesser extent to the shift category. This reflects the fact that the SPE system can improve the pure RBMT translation in terms of better lexical choice and better reordering. Furthermore, the large number of substitutions and insertions in the RBMT system shows that the majority of the errors that account for the lower quality of the RBMT system

System	Ins	Del	Sub	Shift	TER
SMT	5.1	5.05	10.5	3.5	23.92
RBMT	17.04	4.39	30.24	9.3	61.07
SMT+SPE	5.47	4.95	10.1	3.56	24.61
RBMT+SPE	5.2	5.5	10.5	3.27	24.11

Table 4.1: Normalised number of translation errors for the RBMT, SMT, and SPE systems according to TER edit statistics

are lexical. The number of deletions remains largely unaffected by the post-editing system, indicating that little information is actually lost during the second stage. Neither the RBMT+SPE nor the SMT+SPE systems achieve any significant gains over the pure SMT system.

4.3.2 Manual Error Analysis

The manual error analysis involved an in-depth look at the 200 previously extracted sentence, and mapping out the number of errors based on the error typography provided by (Vilar et al., 2006). In a broader sense, the errors can be divided into three groups: lexical, re-ordering and grammatical errors. The "Not Found Words" category represents errors where a word is skipped, i.e. not translated, in the output. "Simple terms" are lexical items, or words, that are mistranslated, and "phrases" represent fragments or sequences of words that are improperly translated. "Meaning" represents errors where the meaning of the original sentence is lost. Determiners, prepositions, tense and number errors are all grammatical errors. Errors that don't fit into one of these categories, but are still grammatical in nature, fall under the "other grammar" category. Word order errors are not grammatical errors, but events where words or phrases occur in the wrong order.

Our error analysis confirms what the TER edit statistics in Table 4.1 suggest, that most of the errors that account for the considerably lower quality of the RBMT

	RBMT	SMT	RBMT+SPE	SMT+SPE
Not Found Words	1.5	5	0	5
Simple Terms	34.5	10.5	6	9.5
Phrases	20.5	2.5	2	3
Meaning	20.5	2.5	2	3
Determiners	1	4.5	2	2.5
Prepositions	3	8.5	2.5	6
Tense	1.5	2.5	2.5	3
Number	0	1	1	1
Other Grammar	2.5	6.5	3.5	5.5
Punctuation	1	3.5	3.5	3.5
Word Order	7	4	4	4.5

Table 4.2: Normalised number and types of errors found in manual evaluation results

system are lexical, both in terms of simple lexical choice and the repercussions of this on the phrasal level. Even though the RBMT system was tuned to the domain of the TM via domain specific lexical resources, most of the errors appear to be due to the RBMT system's inability to pick the right term for the technical domain data set. However, compared to SMT and SMT+SPE, both the RBMT and RBMT+SMT system seem to produce a significantly lower number of grammatical errors, according to our evaluators. This is mostly obvious in the determiner and preposition categories, where combined, the SMT system produces three times as many errors as the RBMT system. Our results also show that while the SPE considerably changes the error typography when applied to RBMT, reducing the overall number of errors, it has has a much smaller effect when applied to the SMT system. SMT+SPE fails to improve on a lexical choice where SMT has failed, and only marginally improved grammatical errors. Example 1 shows a very common RBMT lexical choice error. Errors such as these are almost always corrected in the statistical post-editing (SPE) phase.

Example 1

- *Source*: options de planification de modification d'a pour **un travail de sauvegarde**
- *RBMT*: options of planning of modification of has for **a work of backup**
- *RBMT+SPE*: schedule options to change for **a backup job**
- *SMT*: scheduling options has changed for **a backup job**
- *SMT+SPE*: scheduling options has changed for **a backup job**
- *Reference*: to change schedule options for for **a backup job**

Example 2 shows a similar case where the RBMT+SPE pipleline is superior when it comes to picking the right phrases within the correct domain. Due to the highly technical nature of the Symantec translation memory, the intended meaning is often lost if the wrong lexical choices are made. In this example, **trial version** is the correct technical term for evaluation software. The RBMT system chooses to the phrase: **version of rating**, which is incorrect in this context and loses the intended meaning, despite being a literal translation of **une version d évaluation**.

Example 2

- *Source*: pour installer **une version d évaluation**
- *RBMT*: to install **a version of rating**
- *RBMT+SPE*: to install **a trial version**
- *SMT*: to install **a trial version**
- *SMT+SPE*: to install **a trial version**
- *Reference*: to install **an evaluation version**

On the other hand, RBMT often performs better when it comes to general grammar, especially in terms of prepositions and, to a lesser extent, determiners. This is because RBMT as a translation system focuses primarily on creating sentences that fit a language's rules, focusing on fluency over adequacy. This carries over to the RBMT+SPE system, which leads to a better grammatical quality than the pure SMT system (or the SMT+SPE pipeline, for that matter). Example 3 shows a common case where the preposition is missing from the Moses translation, but is inserted correctly in the RBMT translation. This correct grammar is preserved after post-editing, and carrries over to the RBMT+SPE translations.

Example 3

- *Source*: pour ajouter le **le nom de compte** de connexion
- *RBMT*: to add the name **of account** of login
- *RBMT+SPE*: to add the name **of logon account**
- *SMT*: to add the name **logon account**
- *SMT+SPE*: to add the name **logon account**
- *Reference*: to add the **logon account** name

Another interesting aspect concerns out-of-vocabulary (OOV) words. RBMT seems to be better at finding words than SMT (this is probably a reflection of the fact that the RBMT system used in our experiments was a production system tuned with a domain-specific 10k+ dictionary to the TM-based data-set), and even though these are not always perfectly correct words, they are sometimes fixed in post-editing, as seen in Example 4. As a result, RBMT and RBMT+SPE produce few if any out of vocabulary items in the given output.

Example 4

- *Source*: **enregistrera** l'image .iso idr amorçable ou non amorçable
- *RBMT*: will **record** the or not bootable image .iso idr bootable
- *RBMT+SPE*: will **save** the idr bootable or non-bootable .iso image
- *SMT*: **enregistrera** the idr bootable or non-bootable .iso image
- *SMT+SPE*: **enregistrera** the idr bootable or non-bootable .iso image
- *Reference*: will **save** the bootable or non-bootable idr .iso image .

The results also show that in a few cases SMT+SPE can produce some grammatical improvements over the pure SMT system as well. Example 5 is one such case, where SPE applied to SMT corrected a grammatical error. The preposition **for** is missing in the SMT output and reintroduced by the SPE system in the SMT+SPE output.

Example 5

- *Source*: le nombre de secondes **pendant lesquelles le processus de restauration** ...
- *RBMT*: the number of seconds **during which the process of restoration** ...
- *RBMT+SPE*: the number of seconds **for the restore process** ...
- *SMT*: the number of seconds **the restore process** ...
- *SMT+SPE*: the number of seconds **for the restore process** ...
- *Reference*: the number of seconds **for the restore process** ...

4.4 Summary

In an attempt to gain further understanding of the impact of the SPE system on both the RBMT and the SMT output, we enlisted the help of human evaluators to assess the output of these systems both before and after SPE had been applied. We found that while the automatic metrics seem to favour a pure SMT system, the human annotators were leaning more towards the combined rule based and post-editing system. Furthermore, while automatic evaluation metrics chose Systran as the best system less than 7% of the time, human evaluators chose it as the best system more than twice as often as S-Bleu or Ter did.
We conducted an in-depth error analysis of the same sentences, and found that the SPE system, when applied to RBMT, reduces the overall errors, especially when it comes to lexical choices.

Chapter 5

Quality Estimation for Sentence-Level System Combination

5.1 Introduction

In previous chapters, we relied on a combination of automatic evaluation metrics and manual evaluation to assess the performance of our post-editing systems. The downside of using automatic metrics is that they require a reference translation against which to measure translation quality. As this data, in theory, will not be available at time of testing, we cannot rely on automatic evaluation metrics to estimate the quality of our post-editing system, and whether or not it outperforms our baseline system for specific sentences. Manual evaluation does not face this problem, as human evaluators do not require a reference translation in order to assess the fluency and adequacy of a given sentence. However, manual evaluation is costly and time-consuming, and therefore is an impractical method for on-the-spot evaluation. This is why we turn to quality estimation in order to compare our sentences before and after post-editing. Quality estimation techniques evaluate the quality of output text based on a number of text and system-specific

features, and does not require access to a reference translation.

The use of translation confidence estimation techniques to predict post-editing time and effort has been used previously to some success (Callison-Burch et al., 2012). Most notably, (He et al., 2010a) use confidence estimation techniques to recommend SMT outputs to a Translation Memory user when their classifier predicts that the SMT output is more suitable for post-editing. In a later study, (He et al., 2010b) apply this framework and test it on professional post-editors, reporting that this system can reduce the workload of post-editors. This study can serve as a starting point for our own investigation. While the study by (He et al., 2010a) focuses on predicting suitability for human post-editing, we aim to adapt the techniques and concepts to predict a sentence's suitability for statistical post-editing. At first glance, this seems like a fairly similar task. However, what makes a sentence suitable for a human post-editor may not necessarily apply in statistical post-editing.

Our work in this chapter therefore builds heavily on what we learnt about our post-editing system in Chapter 4. The in-depth manual and automatic analyses of errors that are both corrected and introduced in the post-editing phase serve as a basis for our machine learning task. Like (He et al., 2010a), our aim is to recommend the more suitable sentences to the post-editor.

The remainder of this chapter documents our attempts to use quality estimation techniques to combine the ouput sentences of our systems in order to produce the best output.

5.2 Data Set

The experiments in this chapter use the same translation memory introduced in Section 3.2.1. However, we use the training set described in Section 3.6.3 to extract a new training set for our machine learning system, and a new test set to evaluate our approach. The new training set contains 50,000 sentences. The

sentences are between 1 and 55 words for the English set, and 1 and 76 words for the French set. The average sentence length in the training set is 13 words for English and 15 words for French.

5.3 Support Vector Machines

Amongst previous work on quality estimation, a popular approach is based on features extracted from the source text and its translation, and features specific to the machine translation system, combined with machine learning techniques. As in (He et al., 2010a), we use a support vector machine in order to select our sentences based on which system's output sentence scores highest. For support vector classification, we used LibSVM, a library for Support Vector Machines developed by Chang and Chung (Chang and Lin, 2011). We use two different approaches to choosing better sentences: a classification model, which assigns a binary "better" or "worse" label to a sentence, and a regression model, which estimates a continous score for each sentence.

5.3.1 Classification

Support vector machines classify input based on decision rules which minimize the regularized error function in equation (5.1), which shows a C-SVC algorithm (Cortes and Vapnik, 1995).

$$\begin{aligned} \min_{w,b,\xi} \quad & \frac{1}{2}w^T w + C\sum_{i=1}^{l}\xi_i \\ & y_i(w^T\phi(x_i) + b) \geq 1 - \xi_i, \\ & \xi_i \geq 0, i = 1, \ldots, l. \end{aligned} \tag{5.1}$$

where the training vectors $x_i \in R^n, i = 1, \ldots, l$, are mapped by the function ϕ to a higher dimensional space. w represents the weight vector and ξ represents the

relaxation vector. y is the score we are trying to predict, and b is a constant that needs to be estimated during the training phase. C is the cost parameter, which is optimised with a "gridsearch", which trains the model on different values for C and uses a method of cross-validation to choose the value that achieves the best result. Best results were achieved using the Radial Basis Function (RBF) kernel. The RBF kernel is defined in equation (5.2).

$$K(x_i, x_j) = \exp(-\gamma||x_i - x_j||^2), \gamma > 0 \tag{5.2}$$

where γ is the radius parameter. Like C, γ is optimised using a brute-force gridsearch. The classification result of each set of parameters is evaluated by a 5-fold cross validation on a a 1000 sentence development set.

5.3.2 Regression

Another model frequently used in quality estimation is the regression model. The regression model predicts a continuous score for a sentence. The ϵ-SVR regression algorithm is defined in equation (5.3).

$$\begin{aligned} \min_{w,b,\xi,\xi^*} \quad & \frac{1}{2}w^T w + C\sum_{i=1}^{l}\xi_i + C\sum_{i=1}^{l}\xi_i^* \\ & z_i - w^T\phi(x_i) - b \leq \epsilon + \xi_i, \\ & w^T\phi(x_i) + b - z_i \leq \epsilon + \xi_i^*, \\ & \xi_i, \xi_i^* \geq 0, i = 1, \ldots, l. \end{aligned}$$

where ϵ is the loss function of the ϵ-SVR algorithm, optimised (along with C and γ) through a gridsearch which uses a 5-fold cross-validation method to train the system for different values of ϵ, C and γ.

5.4 Experimental Set-up

5.4.1 Preliminary Work

In Chapter 4 we compared four different machine translation systems: the rule-based system (RBMT), the phrase-based statistical machine translation system (SMT), the rule based system with SPE applied to its output (RBMT+SPE) and the SMT system with SPE applied to its output (SMT+SPE). While some systems drastically outperformed others, each set had individual sentences that outscored the sentences of other systems. The purpose of the experiments described in this section is to attempt to combine the best sentences from each of these outputs.

Tables 5.1 and 5.2 demonstrate this problem by showing us the variation in BLEU scores between sentences that improve with post-editing, and sentences that degrade after post-editing.

	Baseline	PE-NC
EN-FR	61.50	71.54
FR-EN	47.20	57.22

Table 5.1: Comparison of BLEU Scores for a filtered set of sentences that improved after post-editing, comparing SMT and SMT+SPE

	Baseline	PE-NC
EN-FR	61.61	51.55
FR-EN	62.09	52.44

Table 5.2: Comparison of BLEU Scores for a filtered set of sentences that got worse after post-editing, comparing SMT and SMT+SPE

The increase in Bleu scores in Table 5.1 is balanced out by the decrease in Table 5.2, leading to a fairly unchanging Bleu score between the baseline system and

the post-editing system when SPE is applied on the corpus level (i.e. on all the sentences). If we could, however, identify the sentences which improve and post-edit only those, then we could achieve the improvement we are looking for. However, the number of sentences that do change between the SMT and SMT+SPE are very small, accounting for only about 20% of the training set. As these two sets are so similar, and the changes within these sentences so minor, even the oracle score would yield very little overall improvement over the baseline SMT system. For that reason, we chose to simplify our problem and leave out the SMT+SPE system. Since a very small number of RBMT sentences are chosen as the best sentences, we also left out the pure RBMT system.
The Oracle score shows the upper bound, that is, the highest scoring sentences from the baseline and the post-editing system. The oracle is what we hopefully can achieve in a best-case scenario, should we be able to classify these sentences perfectly. The oracle is the evaluation of the test set created from merging the better sentences in the post-edited version with the better sentences in the baseline version.

	SMT	RBMT+SPE	Oracle
FR-EN	65.43	64.63	68.63

Table 5.3: Oracle Scores for the Baseline Postediting System (RBMT+SPE)

As our manual evaluators agreed that the best two systems were the SMT and the RBMT+SPE systems, we decided to investigate the impact of combining the outputs of these two systems at a sentence level. Table 5.3 shows the results of the baseline (SMT) system and the RBMT+SPE, which is the result obtained when applying SPE to the RBMT system in the post-editing phase. The oracle score is the upper bound of both, the theoretical score we could obtain if we used the best of both systems. The oracle score for these two systems gives us an improvement of over 3 BLEU points absolute. This is why we decided to use machine learning methods to choose between the two systems: RBMT+SPE and SMT alone.

5.4.2 Sentence Selection between RBMT+SPE and SMT

We created three different set-ups for our machine learning task. In each set-up, we use TER to compare our output on the sentence level, whether it is to classify our sentences or to score them. Because we use TER throughout our set-up, we will also use TER later for comparison, once we score our new systems.

- Binary Classification: The first one is a binary classification, with 0 and 1 as labels. The labels 1 are for sentences where SMT performs better than the combination system. The labels 0 are for sentences where SMT does not perform better than the RBMT+SPE combination system.
- 3-Way classification: Includes the labels 0, 1 and 2 where
  - 0 indicates a sentence where RBMT+SPE performs better than SMT
  - 1 indicates a sentence where SMT performs better than RBMT+SPE
  - 2 indicates a sentence where the two sentences score the same, despite being different on the surface.
- Regression model: Predicts the difference in TER scores between SMT and RBMT+SPE

In order to fix the problem with the overwhelming bias towards the 0 label for the classification setup, we filtered out all sentences that, on the surface, are entirely identical. The resulting new set consists of 28700 sentence pairs, a little over half of the original set of 50,000 sentences. 27700 of these sentences were randomly chosen to be our training set, and the rest (1000) make up the new test set. Additionally, 1000 sentences from the training set were selected at random to act as a developement set for cross-validation to tune C, γ and ϵ.
Each system considers the input for the pipeline to be the source, so both systems have identical input. Because the systems themselves are different, and we do not have access to the inner workings of Systran (our RBMT system), we treat the

machine translation systems themselves as a black box, and use only system-independent features based on the common source and the different translation outputs.

5.5 Features Overview

A full list of the features can be found in Appendix A.

5.5.1 Baseline Features

Our baseline system consisted of 17 features based on (Specia et al., 2009) and used as a baseline set in recent quality estimation tasks for Workshop on Machine Translation (Callison-Burch et al., 2012). These features are system-independent features extracted from both the source and automatically translated output. The baseline features include surface features such as the number of tokens, the average token length, the number of punctuation marks, and the average number of occurrences of a translated word within the sentence. The features also include n-gram frequencies, language model probabilities, and average number of translations per source word. The full 17 features were extracted using an open source feature extractor developed in Sheffield[1] (Specia et al., 2013). The full set is listed in Appendix A.1. Because the source is identical in both systems, we end up with overlapping features. When we combined these two systems, removing duplicate features, we ended up with a total of 21 features (Appendix A.2). Features were extracted separately for the SMT and RBMT+SPE outputs.

5.5.2 Back-translation Features

In an attempt to improve our prediction accuracy, we included 6 more features, based on the back-translation scores obtained from translating the output of each system back into the source language, using the source as a reference translation

[1]https://github.com/lspecia/quest

and the output itself as the source. We use the same SMT system, trained on different source data, but the same reference data (the original source) to translate these sentences back into the source language. We scored these source sentences using S-BLEU, TER and Fuzzy Match Scores based on the Levenshtein Distance (Levenshtein, 1966) as described in equation (5.3).

$$FuzzyMatch(t) = 1 - \min_{e} \frac{LevenshteinDistance(s, e)}{Len(s)} \tag{5.3}$$

The resulting 6 features were added to the original 21 features to produce our extended 27 features.

5.5.3 Ter Edit Statistic Features

In addition to the 6 back-translation features, we extracted TER edit statistics based on the difference between the SMT and RBMT+SPE outputs as previously used in (Okita et al., 2012). These features were extracted for each sentence by treating the SMT output as a "hypothesis" translation, and comparing it to the RBMT+SPE output, which is being treated as a "reference" translation. This results in five different features that represent the number of insertions, deletions, substitutions, shifts and word shifts between the SMT output and the RBMT+SPE output. Additionally, we also used the TER score itself as a feature.

5.5.4 Part of Speech (PoS) Features

We used the Stanford POS tagger to tag both the English and French side of the translation memory, as extracted in (He et al., 2010a). The Stanford Tagger uses both the preceding and following tag contexts, in addition to lexical features, to tag text. The Stanford Tagger achieves a 97.24% accuracy on the Penn Treebank WSJ (Toutanova et al., 2003). After tagging the source (French), the SPE output (English) and the SMT output (English), we counted up all the tags and assigned them different categories outlined in Table 5.4.

Category	French Tags	English Tags
verbs	V	V VB VBZ VBN VBG
nouns	N	NN NNS
pronouns	CL	PRN
prepositions	P	PRP
determiners	D	DT
conjunctions	C	CC
adverbs/adjectives	A	JJ

Table 5.4: Conflation of English and French PoS tags

For each category, we calculated a ratio between the source and the two hypothesis sentences by simply dividing the number of times a specific category is covered in the source sentence by the number of times the same category is covered in the hypothesis sentence. In situations where the category is not covered in the hypothesis sentence, but is covered in the source, we set this feature to "-1". This resulted in 14 additional features which we added to our set.

5.5.5 Experimental Results

We report three sets of results depending on the classification type: the binary classification, the 3-way classification, and the regression model. In all of these set-ups, we used the RBF kernel and optimise the SVM parametres by maximising the classification accuracy for the classification tasks and minimising the regression errors for the regression tasks.

	Binary	3-way
Accuracy	56.87%	49.55%

Table 5.5: Accuracy results for binary and 3-way classification

The 3-way classifier achieves a lower accuracy than the binary classifier, which

given the 3-way classification is an overall more difficult task, is expected. Both classifiers fail to achieve satisfactory results, and when applied to the test set, failed to improve the overall score over that of the SMT system output.

Finally, we applied the results of our classifiers to the sentences in each of the test sets. In the case of the binary classifier (which we chose at it achieves a higher accuracy over the 3-way classifier), we created a new test set with the "better" sentences chosen by the classifier. If the classifier labelled a sentence as 0, we added the RBMT+SPE output to the new test set. If the classifier labelled it as 1 we added the SMT output. We then scored the new test set using TER, given that we used TER to classify them in the first place. The test set selected by the binary classifier did not improve over the baseline SMT system (29.78), nor did it significantly improve over the RBMT+SPE system (31.55).

Similarly, we applied the regression model's choices to the same test set. The predicted scores of the regression model is the difference between the SMT output's TER score and the RBMT+SPE output's TER score. Therefore, if the predicted score is negative, then the SMT score is considered lower (and therefore better, as in TER, lower scores express a better score), and that system's output sentence is added to the test set. If the predicted score is positive, then the RBMT+SPE score is considered lower and that system's output sentence is added instead. If the predicted score is 0, the SMT score is added. The chosen sentences are then scored using TER on a document level. Judging from TER scores, the regression model seems to perform better at choosing the best sentences than the classification model.

The results of the new sets chosen by both the binary classifier and the regression model are summarised in Table 5.6.

The improvement in TER from 29.78 (SMT system) to 29.16 (in the selected sentences) shows that we are able to use a regression model to select the better sentences. Using approximate randomisation, we determined that this change is statistically significant. Approximate randomisation samples permutations in order

	SMT	RBMT+SPE	Selected Set (binary)	Selected Set(regression)	oracle
TER	29.78	31.55	31.13	29.16	25.94

Table 5.6: New selected test set scored with TER and compared to each of the systems

to approximate a paired permutation test (Noreen, 1989). By randomly exchanging sentences between the two systems we are comparing, we can measure the probability that the difference in TER scores arose by chance. The probability p was determined to be 0.02 using Multeval[2] (Clark et al., 2011). However, the difference in BLEU and METEOR scores is not statistically significant. This shows that the system is heavily biased towards the specific metric we used to score our training set, in this case TER.

5.6 Summary

In this chapter, we documented our attempts to use quality estimation techniques to select the better system's output on a sentence level. We chose to select between the SMT and the RBMT+SPE systems, because of the promising oracle score between the two outputs. We built three types of classifiers: a binary classifier, a 3-way classifier, and a regression model using a set of 47 system-independent features. The features are divided into three separate sets: the 17 baseline features for each system (21), the back-translation features (6), the edit statistics (6), and the part of speech information (14). A full list of these features is available in Appendix A.

Our results showed that the regression model achieves the highest accuracy and is able to select sentences to a degree that improves our test set's overall TER score from 29.78 (SMT system) to 29.16.

[2]https://github.com/jhclark/multeval

Chapter 6

Conclusions and Future Work

In this thesis, we explored the applications of a statistical post-editing system trained on the output of two different machine translation systems, a rule based system and a phrase-based statistical machine translation system. We investigated the possibility of preserving source context information through a novel context-aware approach, and evaluated our systems using both manual and automatic evaluation metrics. Finally, we attempted to classify our sentences into categories depending on which system provides the best translation for this sentence, in order to select the best possible combination of our output.

Our research begins with Chapter 2, which reviews previous work in both automatic post-editing and quality estimation, and sets the stage for our experiments.

We present our own work starting with Chapter 3, where we describe our Statistical Post-Editing (SPE) system and report our results after applying it to a rule-based and phrase-based statistical machine translation system. While our SPE system appeared to improve over the RBMT system, the "naive" approach to using statistical post-editing (PE) in a pure PB-SMT pipeline does not improve translation for English to French, and only shows a modest improvement for French to English. A novel context-aware approach (PE-CF) with context alignment strength thresholding shows statistically significant improvements of

about 2 BLEU points absolute for all thresholds for the translation direction of French to English. However, further investigation showed that these results were not replicable for different data sets, and therefore cannot be established as a baseline for statistical post-editing.

In Chapter 4, we report the results of the manual evaluation task conducted on a sample of the output in Chapter 3. Our human evaluators all agreed with the automatic evaluation metrics that the RBMT + SPE system does indeed perform better than the RBMT system on its own. However, while they did not find the improvement as pronounced as the automatic evaluation metrics indicate, they consistently rated the RBMT + SPE system higher than the RBMT system by a factor of 2. We also report our detailed error analysis of each of the systems, and found that SPE makes better lexical and phrasal choices, which leads to superior translation quality.

Chapter 5 documents our attempts to use quality estimation methods to predict post-editing difficulty. This task also extended to estimating the quality of each system, using a regression model, in order to successfully select the better sentences from each system output, resulting in a modest improvement of TER scores.

6.1 Research Questions

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

- Can a monolingual second-stage SMT system improve the translation quality of the output?
- How do SMT, RBMT + SPE and SMT + SPE compare when the statistical models are trained on the same data?
- Can we use quality estimation techniques to combine the best sentences outputted by the different post-editing systems?

The question of whether or not a SPE system can be used to improve the translation quality of first-stage systems is addressed in Chapter 3. Our results show that a SPE system applied to RBMT can dramatically improve the output. However, the same system applied to its own output (SMT system output) fails to improve significantly over the baseline. In addition we raised the question of whether or not the RBMT + SPE pipeline improves the quality of the output over that of the pure SMT system, trained on the same data set. Chapter 4 tackles this question. At first glance, automatic evaluation metrics seemed to show that SPE does not improve RBMT systems over the pure SMT system. However, a manual evaluation showed that human translators do prefer the RBMT + SPE output over the pure SMT output. We conclude that this discrepancy is a result of BLEU and TER being biased towards the SMT system. Furthermore, error analysis shows that SPE makes better lexical and phrasal choices, which leads to superior translation quality. Our final research question deals with machine learning methods and selecting the best machine translation output. We chose to combine the outputs from the RBMT+SPE and the SMT systems. We left out the SMT+SPE system as it is largely similar to the SMT system on its own, and we left out the RBMT system as it very rarely outputs the best sentence. Chapter 5 attempts to tackle this problem, and shows that a regression model trained on a set of surface features, backtranslation features, TER edit statistics, and part of speech features manages to estimate the difference in translation quality to a limited degree. Using this model to select our best sentences from each system, we managed to improve the TER score by 0.62 absolute. Although so far the performance of the classification system has been less than promising, we would need to perform more experiments on a larger range of features before we can rule it out entirely.

6.2 Future Work

Over the course of this research, we have identified a number of statistical post-editing problems and investigated solutions. The main problem, which addresses the errors introduced by the post-editing system, is addressed in Chapter 5, where we attempt to select the more correct sentences from the post-edited system and the pure SMT system. While we have only managed to achieve modest improvements from sentence selection, we have identified some avenues of future work which might further address this problem.

There are several different approaches to quality estimation, and while we have focused on the most advanced and novel approaches, we have not fully investigated the possibility of combining our systems on a phrase level rather than a sentence level. Previous work into phrase-based quality estimation has proven successful when used to guide system combination (Okita et al. (2012)).

While our source context-modelling approach introduced in Chapter 3 was not successful in improving our baseline post-editing system, source context information might still be useful for post-editing and sentence classification. New language-model based features using the source context language introduced in Section 3.4.1 might help improve the accuracy of quality estimation. Furthermore, we have so far treated our translation systems as a black box, and used only system-independent features in our attempts to estimate translation quality for sentence classification. Using system-dependent features might help improve the accuracy and the overall quality of the combined output.

Appendix A

A.1 Baseline Feature Set (17 Features)

1. number of tokens in the source sentence

2. number of tokens in the target sentence

3. average source token length

4. LM probability of source sentence

5. LM probability of the target sentence

6. average number of occurrences of the target word within the target sentence

7. average number of translations per source word in the sentence (as given by IBM 1 table thresholded so that prob(t|s) > 0.2)

8. average number of translations per source word in the sentence weighted by the inverse frequency of each word in the source corpus

9. percentage of unigrams in quartile 1 of frequency (lower frequency words) in a corpus of the source language

10. percentage of unigrams in quartile 4 of frequency (higher frequency words) in a corpus of the source language

11. percentage of bigrams in quartile 1 of frequency of source words in a corpus of the source language

12. percentage of bigrams in quartile 4 of frequency of source words in a corpus of the source language

13. percentage of trigrams in quartile 1 of frequency of source words in a corpus of the source language

14. percentage of trigrams in quartile 4 of frequency of source words in a corpus of the source language

15. percentage of unigrams in the source sentence seen in a corpus (SMT training corpus)

16. number of punctuation marks in source sentence

17. number of punctuation marks in target sentence

A.2 Combined Feature Set (21 Features)

1. number of tokens in the source sentence

2. number of tokens in the target sentence (moses)

3. average source token length

4. LM probability of source sentence

5. LM probability of the target sentence (moses)

6. average number of occurrences of the target word within the target sentence (moses)

7. average number of translations per source word in the sentence (as given by IBM 1 table thresholded so that prob(t|s) > 0.2)

8. average number of translations per source word in the sentence weighted by the inverse frequency of each word in the source corpus

9. percentage of unigrams in quartile 1 of frequency (lower frequency words) in a corpus of the source language

10. percentage of unigrams in quartile 4 of frequency (higher frequency words) in a corpus of the source language

11. percentage of bigrams in quartile 1 of frequency of source words in a corpus of the source language

12. percentage of bigrams in quartile 4 of frequency of source words in a corpus of the source language

13. percentage of trigrams in quartile 1 of frequency of source words in a corpus of the source language

14. percentage of trigrams in quartile 4 of frequency of source words in a corpus of the source language

15. percentage of unigrams in the source sentence seen in a corpus (SMT training corpus)

16. number of punctuation marks in source sentence

17. number of punctuation marks in target sentence (moses)

18. number of tokens in the target sentence (SPE)

19. LM probability of the target sentence (SPE)

20. average number of occurrences of the target word within the target sentence (SPE)

21. number of punctuation marks in target sentence (SPE)

A.3 Extended Feature Set

1. number of tokens in the source sentence

2. number of tokens in the target sentence (moses)

3. average source token length

4. LM probability of source sentence

5. LM probability of the target sentence (moses)

6. average number of occurrences of the target word within the target sentence (moses)

7. average number of translations per source word in the sentence (as given by IBM 1 table thresholded so that prob(t|s) 0.2)

8. average number of translations per source word in the sentence

weighted by the inverse frequency of each word in the source corpus

9. percentage of unigrams in quartile 1 of frequency (lower frequency words) in a corpus of the source language

10. percentage of unigrams in quartile 4 of frequency (higher frequency words) in a corpus of the source language

11. percentage of bigrams in quartile 1 of frequency of source words in a corpus of the source language

12. percentage of bigrams in quartile 4 of frequency of source words in a corpus of the source language

13. percentage of trigrams in quartile 1 of frequency of source words in a corpus of the source language

14. percentage of trigrams in quartile 4 of frequency of source words in a corpus of the source language

15. percentage of unigrams in the source sentence seen in a corpus (SMT training corpus)

16. number of punctuation marks in source sentence

17. number of punctuation marks in target sentence (moses)

18. number of tokens in the target sentence (SPE)

19. LM probability of the target sentence (SPE)

20. average number of occurrences of the target word within the target sentence (SPE)

21. number of punctuation marks in target sentence (SPE)

22. Moses backtranslation scored with BLEU

23. Moses backtranslation scored with TER

24. Moses backtranslation scored with Levenshtein

25. Systran+Moses backtranslation scored with BLEU

26. Systran+Moses backtranslation scored with TER

27. Systran+Moses backtranslation scored with Levenshtein

28. Number of insertions between SPE and SMT

29. Number of deletions between SPE and SMT
30. Number of substitutions between SPE and SMT
31. Number of shifts between SPE and SMT
32. Number of word shifts between SPE and SMT
33. TER score between SPR and SMT
34. Ratio of verbs from source to target (SMT)
35. Ratio of nouns from source to target (SMT)
36. Ratio of adjectives/adverbs from source to target (SMT)
37. Ratio of prepositions from source to target (SMT)
38. Ratio of pronouns from source to target (SMT)
39. Ratio of determiners from source to target (SMT)
40. Ratio of conjunctions from source to target (SMT)
41. Ratio of verbs from source to target (SPE)
42. Ratio of nouns from source to target (SPE)
43. Ratio of adjectives/adverbs from source to target (SPE)
44. Ratio of prepositions from source to target (SPE)
45. Ratio of pronouns from source to target (SPE)
46. Ratio of determiners from source to target (SPE)
47. Ratio of conjunctions from source to target (SPE)

Bibliography

Allen, J. and Hogan, C. (2000). Toward the Development of a Post editing Module for Raw Machine Translation Output: A Controlled Language Perspective. In *Proceedings of the Workshop on Controlled Language Applications (CLAW)*, pages 62–71.

Béchara, H., Ma, Y., and van Genabith, J. (2011). Statistical Post-Editing for a Statistical MT System. In *Proceedings of the MT Summit XIII*, pages 308–315.

Béchara, H., Rubino, R., Ma, Y., and van Genabith, J. (2012). An Evaluation of Statistical Post-editing Systems Applied to RBMT and SMT Systems. In *Proceedings of the 24th International Conference on Computational Linguistics (CoLing2012), Mumbai, India*.

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

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

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

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

Chang, C.-C. and Lin, C.-J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27.

Clark, J., Dyer, C., Lavie, A., and Smith, N. (2011). Better Hypothesis Testing for Statistical Machine Translation: Controlling for Optimizer Instability. In *Association for Computational Lingustics (ACL)*.

Cortes, C. and Vapnik, V. (1995). Support-Vector Networks. *Machine Learning*, 20:273–297.

Dugast, L., Senellart, J., and Koehn, P. (2007). Statistical Post-Editing of SYSTRAN's Rule-Based Translation System. In *Proceedings of the Second Workshop on Statistical Machine Translation (WMT)*, pages 220–223.

Gandrabur, S. and Foster, G. (2003). Confidence estimation for text prediction. In *Proceedings of the 7th Conference on Natural Language Learning (CoNLL), Edmonton, Alberta,*.

Guillevic, D., Gandrabur, S., and Normandin, Y. (2002). Robust semantic confidence scoring. In *Proceedings of the 7th International Conference on Spoken Language Processing (ICSLP) 2002, Denver, Colorado*.

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

He, Y., Ma, Y., van Genabith, J., and Way, A. (2010b). Improving the post-editing experience using translation recommendation: a user study. In

Proceedings of the the ninth conference of the association for Machine Translation in the Americas, pages 247–256.

Isabelle, P., Simard, M., and Goutte, C. (2007). Domain Adaptation of MT Systems through Automatic Post-editing. In *Proceedings of MT Summit X-1 Copenhagen, Denmark*, pages 225–261.

Kneser, R. and Ney, H. (1995). Improved backing-off for m-gram language modeling. In *In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, volume I*, pages 181–184.

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

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

Kuhn, R., Isabelle, P., Goutte, C., Senellart, J., Simard, M., and Ueffing, N. (2010). Automatic Post-Editing. *Multilingual*, 21(1):43–46.

Lagarda, A. L., Alabau, V. V., Casacuberta, F., Silva, R., and Diaz-de Liano, E. (2009). StatisticalPost-Editing of a Rule-Based Machine Translation System. In *Proceedings of NAACL HLT*, volume Short Papers, pages 217–220.

Landis, J. and Koch, G. (1977). The measurement of observer agreement for categorical data. *Biometrics*, pages 159–174.

Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, pages 707–710.

Lin, C.-Y. and Och, F. J. (2004). A Method for Evaluating Automatic Evaluation Metrics for Machine Translation. In *Proceedings of the 20th international conference on Computational Linguistics*.

Ma, C., Randolph, M., and Drish, J. (2002). A support vector machines-based rejection technique for speech recognition. In *ICASSP 2001*.

Moreno, P., Logan, B., and Raj, B. (2001). A boosting approach for confidence scoring. In *Eurospeech*.

Noreen, E. (1989). Computer-Intensive Methods for Testing Hypotheses: An Introduction. In *Wiley-Interscience*.

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

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

Oflazer, K. and El-Khalout, I. D. (2007). Exploring Different Representational Units in English-to-Turkish Statistical Machine Translation. In *Proceedings of the Second Workshop on Statistical Machine Translation (WMT)*, pages 25–32.

Okita, T., Rubino, R., and van Genabith, J. (2012). Sentence-Level Quality Estimation for MT. In *Proceedings of the 24th International Conference on Computational Linguistics (CoLing2012), Mumbai, India*.

Papineni, K., Roukos, S., Ward, T., and Zhu, W. (2002). BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the Association for Computational Linguistics (ACL)*, pages 311–318.

Potet, M., Besacier, L. Blanchon, H., and M, A. (2012). Towards a Better Understanding of Statistical Post-Edition Usefulness. In *Proceedings of the 9th International Workshop on Spoken Language Translation (IWSLT)*.

Potet, M., Esperança-Rodier, E., Blanchon, H., and Besacier, L. (2011). Preliminary experiments on using users post-editions to enhance a smt system. In *Proceedings of the 15th EAMT*, pages 161–168.

Roturier, J. (2009). Deploying Novel MT Technology to Raise the Bar for Quality: A Review of Key Advantages and Challenges. In *Proceedings of the MT Summit XII*.

Rubino, R., Huet, S., Lefèvre, F., and Lenarés, G. (2012). Statistical Post-Editing of Machine Translation for Domain Adaptation. In *Proceedings of the European Association for Machine Translation (EAMT)*, pages 221–228.

Sanchis, A., Juan, A., and Vidal, E. (2003). Improving utterance verificationusing a smoothed naive Bayes model. In *ICASSP 2003*.

Simard, M., Goutte, C., and Isabelle, P. (2007). Statistical Phrase-based Post-editing. In *Proceedings of the Human Language Technology Conference and the North American Chapter of the Association for Computational Linguistics (HLT/NAACL)*, pages 508–515.

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

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

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

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

Steinberger, R., Pouliquen, B., Widiger, A., Ignat, C., Erjavec, T., and Tufis, D. (2006). The JRC-Acquis: A multilingual aligned parallel corpus with 20+ languages. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC–2006*, pages 2142–2147.

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

Terumasa, E. (2007). Rule Based Machine Translation Combined with Statistical Post Editor for Japanese to English Patent Translation. In *Proceedings of the MT Summit XI Workshop on Patent Translation*, volume 11, pages 13–18.

Thurmair, G. (2004). Comparing rule-based and statistical MT output. In *Workshop on the amazing utility of parallel and comparable corpora (2004)*, pages 5–9.

Toutanova, K., Manning, C., Klein, D., and Singer, Y. (2003). Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. In *Proceedings of HLT-NAACL*, pages 252–259.

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

Vilar, D., Xu, J., D'Haro, L. F., and Ney, H. (2006). Error Analysis of Statistical Machine Translation Output. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC)*, pages 697–702.

Wang, Y.-Y. (1998). Grammar inference and statistical machine translation.