# The Role of Syntax and Semantics in Machine Translation and Quality Estimation of Machine-translated User-generated Content

Rasoul Samad Zadeh Kaljahi

B.Eng., M.Sc.

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

Doctor of Philosophy (Ph.D.)

to the


Dublin City University
School of Computing

Supervisors:
Dr. Jennifer Foster
Dr. Johann Roturier (Symantec, Ireland)

February 2015

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Ph.D. 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.:

Date:

# Contents

# List of Figures

# List of Tables

# Abbreviations

| | |
|---|---|
| **BLEU** | Bilingual Evaluation Understudy |
| **CFG** | Context-free Grammar |
| **ETTB** | English Translation Treebank |
| **EWTB** | English Web Treebank |
| **FTB** | French Treebank |
| **GTM** | General Text Matcher |
| **HBLEU** | Human-targeted Bilingual Evaluation Understudy |
| **HMeteor** | Human-targeted Meteor |
| **HP** | Hierarchical Phrase-based Machine Translation |
| **HTER** | Human-targeted Translation Error Rate |
| **LAS** | Labelled Attachment Score |
| **LTPAM** | Lexical Translation Table-based Predicate-Argument Structure Match |
| **MERT** | Minimum Error Rate Tuning |
| **MT** | Machine Translation |
| **NANC** | North American News Text Corpus |
| **NLP** | Natural Language Processing |
| **OOV** | Out-of-vocabulary |
| **PAM** | Predicate-Argument Structure Match |
| **PB** | Phrase-based Machine Translation |
| **PCFG** | Probabilistic Context-Free Grammars |
| **PCFG-LA** | Probabilistic Context-Free Grammars with Latent Annotations |
| **POS** | Part-of-speech |
| **PTB** | Penn Treebank |
| **PTPAM** | Phrase Translation Table-based Predicate-Argument Structure Match |
| **QE** | Quality Estimation |
| **RMSE** | Root Mean Square Error |
| **SMT** | Statistical Machine Translation |
| **SRL** | Semantic Role Labelling |
| **ST** | String-to-tree Machine Translation |
| **SVM** | Support Vector Machine |
| **TER** | Translation Error Rate |
| **TS** | Tree-to-string Machine Translation |
| **TT** | Tree-to-tree Machine Translation |
| **UAS** | Unlabelled Attachment Score |
| **UGC** | User-generated Content |
| **WAPAM** | Word Alignment-based Predicate-Argument Structure Match |
| **WSJ** | Wall Street Journal |

# The Role of Syntax and Semantics in Machine Translation and Quality Estimation of Machine-translated User-generated Content

Rasoul Samad Zadeh Kaljahi

## Abstract

The availability of the Internet has led to a steady increase in the volume of online user-generated content, the majority of which is in English. Machine-translating this content to other languages can help disseminate the information contained in it to a broader audience. However, reliably publishing these translations requires a prior estimate of their quality. This thesis is concerned with the statistical machine translation of Symantec's Norton forum content, focusing in particular on its quality estimation (QE) using syntactic and semantic information.

We compare the output of phrase-based and syntax-based English-to-French and English-to-German machine translation (MT) systems automatically and manually, and find that the syntax-based methods do not necessarily handle grammar-related phenomena in translation better than the phrase-based methods. Although these systems generate sufficiently different outputs, the apparent lack of a systematic difference between these outputs impedes its utilisation in a combination framework.

To investigate the role of syntax and semantics in quality estimation of machine translation, we create *SymForum*, a data set containing French machine translations of English sentences from Norton forum content, their post-edits and their adequacy and fluency scores. We use syntax in quality estimation via tree kernels, hand-crafted features and their combination, and find it useful both alone and in combination with surface-driven features. Our analyses show that neither the accuracy of the syntactic parses used by these systems nor the parsing quality of the MT output affect QE performance. We also find that adding more structure to French Treebank parse trees can be useful for syntax-based QE.

We use semantic role labelling (SRL) for our semantic-based QE experiments. We experiment with the limited resources that are available for French and find that a small manually annotated training set is substantially more useful than a much larger artificially created set. We use SRL in quality estimation using tree kernels, hand-crafted features and their combination. Additionally, we introduce *PAM*, a QE metric based on the predicate-argument structure match between source and target. We find that the SRL quality, especially on the target side, is the major factor negatively affecting the performance of the semantic-based QE.

Finally, we annotate English and French Norton forum sentences with their phrase structure syntax using an annotation strategy adapted for user-generated text. We find that user errors occur in only a small fraction of the data, but their correction does improve parsing performance. These treebanks (*Foreebank*) prove to be useful as supplementary training data in adapting the parsers to the forum text. The improved parses ultimately increase the performance of the semantic-based QE. However, a reliable semantic-based QE system requires further improvements in the quality of the underlying semantic role labelling.

# Acknowledgments

This thesis would not have been possible without the valuable support of many people. I would like to take this opportunity to express my gratitude for all the valuable help I received from them.

I have been very lucky to be involved in the ConfidentMT project, where I met and was supervised by a team of experienced researchers and wonderful people. First and foremost, I would like to thank my supervisor, Dr. Jennifer Foster, for her outstanding supervision. The amount of support and help I received from her can not be expressed in words. Thank you Jennifer for your patience, understanding and kindness. I would also like to thank Dr. Johann Roturier, my industrial supervisor, for his valuable advices and cooperation. He always helped me to continue on the right track. I am also grateful to Dr. Fred Hollowood, who mentored me throughout the PhD work and supported the creation of two invaluable data sets, SymForum and Foreebank, for my studies and for the research community. My special thanks goes to Dr. Raphael Rubino for his valuable comments on my work, the things I learned from him and the encouragements he gave me when I needed. And my special thanks to Dr. Matthew Elder for being so kind and extremely supportive during his direction of our research group at Symantec Research Labs.

I am thankful to Irish Research Council and Symantec Ireland for co-funding my study through the ConfidentMT project. I would also like to thank CNGL (now ADAPT) for the infrastructure they provided in the PhD lab and for supporting my PhD at its final stages. Many thanks to Dr. Joachim Wagner, who always provided help and support for using the CNGL cluster with patience.

I would also like to express my gratitudes to my examiners, Dr. Lluís Màrquez and Prof. Qun Liu. Their insightful and constructive comments helped improve this thesis further. I am also grateful to Prof. Andy Way, the chair of my defence session, who made sure that everything went smoothly during the viva. Special thanks to Prof. Josef van Genabith for two things; first for connecting me to the ConfidenMT project and second for his valuable comments as the examiner of my transfer talk. I also thank Dr. Alistair Sutherland for providing useful suggestions as the second transfer report examiner.

In these three years of study, I spent my time working beside many amazing people at the CNGL/NCLT lab in DCU and at the Symantec Research Labs. I learned many things from them during our enjoyable and refreshing lunchtime chats and discussions and I received enormous help and inspiration from them. Thank you all guys and apologies that I cannot name all of you one by one.

And now is the moment to thank the one who never left me alone, even a day, during these long (though short) years, and always gave me inspiration and encouragement. I am indebted to my fiancée, Shabnam, who will soon be my wife; thank you a universe for all your patience and sacrifices. I would like to thank my family who always offered me their unconditional support and who were such amazing father, mother and brother; I would never be here without you. I am grateful to my father, mother, brother and sister in law for their kindness, understanding and patience, and for raising such a wonderful daughter. Many thanks to my uncles and aunties who never forgot me and always offered me their sincere help and support.

Last but not least, I would like to thank all my life friends for their faithful friendship and for all the encouragements and supports I received from them. Special thanks to Mustafa Hijrat, my best friend, who was the one who helped me shape the decision of continuing graduate studies after years of working in industry; the right decision for me!

# Chapter 1

# Introduction

As the popularity and availability of the Internet, and the Web in particular, increases, more and more people are contributing to the ever-growing body of online content. Newsgroups, discussion forums, social networks, weblogs, microblogs and consumer reviews are examples of online media, which contain content generated by users, commonly known as *user-generated content* or *UGC*. For instance, the customer service model is moving away from the traditional way of providing help via phone lines to one in which customers help each other via *forums*. The work presented in this thesis takes place in the context of a wider project, *ConfidentMT*, the ultimate goal of which is to improve and estimate the quality of machine translation of user-generated content from Symantec Norton forums,[1] where Norton users, including Symantec technical support employees, discuss the Norton security products. Although Norton forums exist for multiple languages including English, French and German, it contains far more content in English than the other languages. Reliable machine translation of the English content to French and German, as the next most popular languages of the Norton forums, can help disseminate the potentially valuable information to more users. However, in order for these translations to be published, an estimate of their reliability is required. Figure 1.1 presents an example of a machine translation of a sentence from the English Norton forum into French,

[1] `http://community.norton.com`

| | |
|---|---|
| Source | The main 2 websites listed below: |
| Machine translation | La principale 2 de sites Web répertoriés ci-dessous: |
| Corrected translation | Les 2 principaux sites Web répertoriés ci-dessous: |

Figure 1.1: Example of machine translation of an English Norton forum sentence containing error (missing *are*) to French

where the missing *are* in the source has led to an incorrect translation, which is not suitable for publishing. To this end, the project targets two problems: 1) improving the quality of machine translation to achieve reliable translations and 2) automatically estimating the quality of these translations, known as *quality estimation* or *QE*, to measure this reliability. This thesis focuses on addressing these problems from a more specific perspective. In particular, it investigates the use of syntactic knowledge in statistical machine translation of the Norton forum text and both syntactic and semantic knowledge in its quality estimation. The aim of using syntax and semantics in these tasks is to employ a deeper level of analysis of the text and its translations rather than relying on shallow surface-driven information.

In the rest of this chapter, we first introduce each of these problems, the motivation behind the chosen approaches as well as the fundamental research questions addressed in this thesis. In addition, we provide an overview of our methodological approaches and the findings of the experiments. Section 1.1 is dedicated to machine translation and Section 1.2 to quality estimation. In Section 1.3, we introduce our study on the syntax of Norton forum text. Section 1.4 presents the structure of the thesis. Finally, Section 1.5 lists the publications resulting from this research.

## 1.1 Machine Translation

Machine translation was introduced in the middle of the last century (Pierce and Carroll, 1966). The early machine translation systems were based on linguistic analysis of the source and target languages, where rules were developed to map the morphological, syntactic and semantic structures of the source to the target language. The extraction of these rules was labour-intensive and required linguis-

tic expertise, limiting their coverage. These shortcomings led to the emergence of data-driven machine translation, in which such rules are extracted from parallel corpora. Statistical machine translation (SMT) is the most well established data-driven translation paradigm, which emerged with the introduction of the *IBM Models* by Brown et al. (1988) and has since been extensively studied. The IBM Models are based on the alignment between the source and target words in a parallel corpora and ignore the syntactic structure of both languages. Current statistical machine translation methods extend these models to the phrase level. The *alignment template* model proposed by Och et al. (1999) takes the word context into account and handles local word order in translation by aligning sequences of adjacent words, i.e. shallow phrases, rather than single words. Koehn et al. (2003) show that translation with even short phrases of three words outperforms word-based translation. However, such phrases are merely sequences of strings and do not necessarily represent linguistic structures. Several attempts have been made to design methods which incorporate syntactic knowledge into statistical machine translation using various approaches such as syntactic noisy channel translation (Yamada and Knight, 2001), syntactic transformation rule extraction (Galley et al., 2004), dependency treelet translation (Xiong et al., 2007), forest-based translation (Mi et al., 2008) and, more recently, syntax as a soft constraint (Chiang, 2010; Zhang et al., 2011). These *syntax-based* SMT methods allow the structural mapping between languages to be captured. Once captured, such differences can account for phenomena such as long-distance reordering which are known to be a deficit of the ad-hoc phrase-based models.

Despite all these efforts, studies show that syntax-based machine translation methods do not necessarily work better. One of the fundamental factors affecting the performance of such methods appears to be the language pair to which they are applied. It has been suggested that structurally distant language pairs tend to benefit from syntax-based methods while similar languages are better translated with phrase-based methods (Galley et al., 2004; Zollmann and Venugopal, 2006). In this thesis we bring syntax-based SMT under closer scrutiny, in order to understand

its performance on the target language pairs and data of the project, i.e. English-French and English-German translation of the Norton forum text (Chapter 2). Not only does this comparison shed light on the utility of syntax in machine translation, finding such differences can help in combining all these various approaches to benefit from the advantages of each of them. Using French and German, two different target languages in terms of their structural similarity to English, will better help understand the effect of this factor on the performance of syntax-based methods. The next section explains the research questions we specifically try to address here.

### 1.1.1 Research Questions

We systematically look at the differences between phrase-based and syntax-based SMT methods in translating forum content by analysing their output both automatically and manually. We seek to answer the following questions:

- *How different are the outputs generated by each of these methods?*
- *Can the outputs be beneficially combined in theory?*
- *Are any differences between the two types of systems systematic enough to be exploited in system combination?*

The next section summarises the experiments carried out to find answers for these questions as well as the findings.

### 1.1.2 Summary and Findings

We investigate the use of syntax in statistical machine translation by comparing phrase-based and syntax-based machine translation methods. The phrase-based approaches include regular (called phrase-based henceforth) and hierarchical phrase-based and the syntax-based approaches include tree-to-string, string-to-tree and tree-to-tree methods all implemented in the Moses machine translation toolkit (Hoang et al., 2009). All the translation and language models are trained on the data from Symantec translation memories which contain a mixture of Symantec content from

product manuals, software strings, marketing materials, knowledge bases and websites translated from English to French and German. The performance of syntax-based systems may be influenced by the quality of the underlying automatic syntactic analysis (Quirk and Corston-Oliver, 2006). When applied to the user-generated Norton forum text, the problem can be further exacerbated by the fact that commonly used statistical syntactic parsers are trained on edited newswire text and do not generalize well to unedited text from a different domain (Foster et al., 2011a). To account for this factor in our analysis, we evaluate the SMT approaches under comparison on both well-formed content drawn from Symantec software documentation and user-generated content from Norton forums.

The systems are compared in both an automatic and manual manner. The automatic comparison involves comparing the outputs produced by the translation systems to discover whether different methods generate different outputs for the same sentences, so that the best of each can be exploited by combining them. Several methods based on evaluation using automatic metrics such as BLEU and TER are used at both the document and sentence level. We find noticeable differences among the outputs of different methods and their oracle combination proves to be significantly fruitful. We extend the automatic comparison of systems using a manual evaluation procedure, to discover which system is good at handling which translation phenomena. We find that the syntax-based systems do not particularly produce a more grammatical output in terms of, for example, agreement or word order. Our findings show that despite a significant potential to improve the translation quality through the combination of these methods, a systematic difference between them cannot be found.

## 1.2 Quality Estimation

Regardless of the method used for machine translation of forum text, publishing the translated text requires confidence in the quality of those translations. How-

ever, evaluating these translations manually is costly, time-consuming and perhaps in contrast to the motivation of machine translation which is to reduce these costs (Roturier and Bensadoun, 2011). Therefore, the ability to automatically estimate the quality of these translations will assist in confidently publishing the translations. In addition, this estimation can provide a means to select the best of the translations produced by different systems, i.e. to combine machine translation systems, which is also envisioned in the comparison of SMT methods described in the previous section. The performance of current quality estimation methods, however, is not sufficient for this application. While machine translation has consistently been receiving considerable attention, we find a bigger niche in quality estimation research and application. Inspired by this need, the main focus of this thesis is on quality estimation. We investigate methods in quality estimation (QE) systems for machine translation. For this purpose, we create a data set containing sentences selected from English Norton user forums, their machine translations into French together with the human post-edits of the translations as well as their human evaluation scores (Chapter 3). This data set is used to train and test the QE approaches examined in the thesis.

The quality of machine translation is a multi-faceted concept. Fluency and adequacy are two important aspects, where the former indicates how fluently the translation can be read and the latter how much of the meaning intended by the source utterance it retains. The fluency of a sentence is related to its syntactic construction. A grammatically well-formed sentence is usually read more fluently than its ungrammatical equivalent. Therefore, the syntactic analysis of the translation can provide information useful in estimating its quality. In addition to the translation, the syntactic construction of the source sentence can be predictive of the quality of its translation, by capturing the complexity of the source sentence which is usually correlated with the difficulty of its translation. Besides these characteristics, syntax can also help quality estimation through utilising the syntactic correspondence between the source and target languages. In this thesis, we investigate the use of

syntactic information in quality estimation using various methods (Chapter 4). We additionally conduct an in-depth analysis of various aspects of our syntax-based QE approaches (Chapter 5).

As to the translation adequacy, it can be argued that it is tied to the semantics of both source and its translation. Intuitively, the degree of similarity between the semantic analysis of the source and target can indicate the adequacy of the translation. *Semantic role labelling* (SRL) is a type of shallow semantic analysis which represents the predicate-argument structure of a sentence. Although the semantic role labelling does not fully represent the semantics of the sentence and thus cannot account for its entire meaning, it provides a useful aspect of the meaning by identifying who did what to whom, why, when, where, etc. We experiment with different approaches to using information extracted from the semantic role labelling of the source and its translation in estimating the translation quality (Chapter 7). SRL is chosen as it is well studied and has shown to be useful in various NLP tasks such as MT (Wu and Fung, 2009; Liu and Gildea, 2010), its evaluation (Giménez and Màrquez, 2007; Lo and Wu, 2011) and quality estimation itself (Lo et al., 2014; Pighin and Màrquez, 2011).

Although there are appropriate resources for semantic role labelling of English, French suffers from the lack of adequately sized resources. Only a few researchers have studied French SRL and only a small set of sentences hand-annotated with semantic role labelling currently exists for this language. Alternatively, researchers have tried to use unsupervised machine learning methods or to artificially generate SRL resources using parallel corpora for English and French. We conduct a set of experiments to find the best feasible solution to partially alleviate this problem (Chapter 6).

The following section states the research questions we tackle with regard to each of the studies outlined above.

### 1.2.1 Research Questions

We design and experiment with various methods of using syntactic knowledge in quality estimation in order to verify its usefulness for this purpose and understand the role of various phenomena involved in this process. We specifically seek to answer the following questions:

- *How effective is syntactic information in quality estimation of machine translation both in comparison and in combination with other surface-driven features?*
- *Does parsing accuracy affect the performance of syntax-based QE?*
- *To what extent do the source and target syntax each contribute to the syntax-based QE performance?*
- *Does parsing of noisy machine translation output affect the performance of syntax-based QE?*

Our experiments on semantic role labelling of French aim at finding an approach which can compensate for the lack of resources for this purpose. To accomplish this goal, we try to answer the following questions:

- *How much artificial data is needed to train an SRL system?*
- *Is there a way to improve the projected annotation?*
- *Is a large set of this artificial data better than a small set of hand-annotated data for training a SRL system?*

Similar to the syntax-based QE experiments, we investigate various methods of applying semantic information captured by semantic role labelling in estimating the quality of machine translation and analyse the results to find the problems involved in this approach. We ask the following questions:

- *What is the most effective method of incorporating this semantic knowledge in QE?*
- *To what extent does the semantic predicate-argument structure match between source and target represent the translation quality?*

- *How effective is semantic role labelling, in general, in quality estimation of machine translation both in comparison and in combination with other surface-driven features as well as the syntactic information?*
- *What are the factors hindering the performance of semantic-based QE?*

The next section presents a summary of the quality estimation experiments and their findings.

### 1.2.2 Summary and Findings

We create two data sets to be used in the quality estimation experiments carried out in this thesis. The first data set, called *SymForum*, is in the target domain of the project and built using sentences from Symantec's English Norton forum machine-translated to French. These machine translations are both post-edited by human translators to obtain human-targeted automatic evaluation scores (Snover et al., 2006) and evaluated to obtain human fluency and adequacy scores. The second data set is *News*, which is built using sentences from the News development data set of WMT 2013 (Bojar et al., 2013). These sentences are machine translated using the same systems as the SymForum sentences (described in Section 3.2.1 of Chapter 3). However, these translations are evaluated using three automatic metrics, BLEU, TER and Meteor, against the available reference translations to generate quality scores. This data set is created in the same domain on which the available syntactic parsers are trained. The purpose of this data set is to factor out the problems resulting from out-of-domain parsing from the syntax-based QE experiments, so that the conclusions made are not affected by such problems. For both data sets, the statistics such as metric correlations and score distribution are extracted. For the SymForum data set, the inter-annotator agreement is calculated and analysed.

Using these data sets, we investigate the use of syntax in quality estimation. We compare two different machine learning methods for incorporating syntactic information in quality estimation: tree kernels (Collins and Duffy, 2002; Moschitti,

2006), an effective method to learn from parse trees which is fast to deploy, and hand-crafted features, a computationally efficient method which offers more design flexibility. The combination of these two methods is also used in building a fully syntactic QE system. With each method, we use both constituency (phrase structure) and dependency parses of the source and target. The performance of the QE systems built using these approaches is compared to a baseline using the surface-oriented baseline features introduced in the WMT 2012 shared task on quality estimation (Callison-Burch et al., 2012). Additionally, the syntax-based systems are combined with this baseline to examine the complementarity between these two types of information. According to the results, syntax-based QE systems significantly outperform the baseline and can also be successfully combined with them for the News data set. However, they are not always better than the baseline when applied to the SymForum data set.

We closely analyse our syntax-based quality estimation methods from different perspectives. Specifically, we examine the effect of parser accuracy on the performance of syntax-based QE systems built upon those parses, as we expect noisy parses for the user-generated and machine translated QE data used here. The parser accuracy is artificially varied by varying the size of the training set of the parser. We find that the syntax-based QE systems are robust to large drops in parsing accuracy. This suggests that, rather than intrinsic measures of parse quality, the intra-document inconsistency of the parses due to inconsistent and noisy language in the forum text may account for the performance gap we observe between the syntax-based QE of the News and the SymForum data set. In addition, we investigate the parts played in syntax-based QE by the syntax of the source and target separately. We find that French constituency parse trees are less useful than the English ones, no matter whether they are extracted from the target (as in the original translation direction) or from the source (as in the reversed translation direction). We also design a set of heuristics which can modify the French parse trees by adding more structure based on the known deficits of the French Treebank (Abeillé et al., 2003)

used to produce these parse trees. The modified trees are significantly more useful than the original ones in the syntax-based QE systems, especially when using the News data set.

We next turn our attention to semantic-based quality estimation, based on the semantic role labelling of the source and target. However, as explained earlier, semantic role labelling of French, as the target language, is challenging since there are limited resources for training a reliable SRL model for this language. We therefore first conduct a series of experiments to build an optimum semantic role labelling system for French with the limited available resources. The experiments are designed based on the idea of projecting the automatic SRL annotation from the English side of a large parallel corpus to its French side using the word alignment between them and using the resulting synthetic data for training a SRL system, as carried out by van der Plas et al. (2011) using the *Europarl* corpus (Koehn, 2005). Attempting to improve the performance of an SRL system trained on the projected annotations, we investigate a variety of methods. These methods include using only direct translation projections between English and French for training and replacing the original POS tags and dependency labels with more coarse-grained universal POS tags and dependency labels. These variations are not shown to be effective. In addition, we compare different word alignments in projecting the annotation. We find that restrictive word alignments aiming at less noisy projected annotations substantially reduce recall and are not preferable over less restrictive ones. We finally compare the use of the large training data obtained by projection with a much smaller set of manually annotated data. It turns out that the latter leads to substantially better performance. Therefore, it is this model which is selected to train the models used to label French data in the semantic-based QE experiments.

With the semantic role labelling of the data in hand, we investigate the use of semantics in quality estimation. Similar to the syntax-based QE systems, we compare the performance of tree kernels and hand-crafted features as well as their combination for this purpose. In addition, we introduce *PAM*, a new metric which

uses the predicate-argument structure match between the source and target as a measure of translation quality. Several variations of this metric are presented differing in the methods used to match the source and target predicates and arguments, including word alignments as well as lexical and phrase translation tables. Manual analysis of the PAM scores shows that the low quality of semantic role labelling is the main factor impeding its accuracy. The PAM scores prove to be more effective when used as features in a machine learning setup and added to the other handcrafted features. Although the semantic-based QE system performs slightly better than the syntax-based system, it is outperformed by the WMT baseline for some settings. We combine it with the syntax-based system and in turn with the baseline features. The best results are obtained when the baseline features are combined with the semantic-based features and semantically augmented tree kernels. However, we believe that a higher quality of semantic role labelling is required in order for these approaches to be useful. It is important for there to be a balance between the quality of the source and target annotation.

## 1.3 The Syntax of Norton Forum Text

The accuracy of the semantic-based quality estimation is dependent on the quality of the semantic role labelling of both source and target, which itself relies on the accuracy of its underlying syntactic analysis as shown by previous studies (Punyakanok et al., 2008). However, it is well known that the syntactic parsers trained on edited newswire resources do not perform well on unedited user-generated data from other domains (Foster et al., 2011a). In order to be able to evaluate the performance of the parsers on the Norton forum text, we build two phrase structure treebanks from this text, named *Foreebank*, one for English and one for French (Chapter 8). The Foreebank annotation strategy accounts for the language and writing errors made by the forum users and for the stylistic conventions of web text. Errors are marked on the parse trees which enables us to analyse user errors and the type of ungrammati-

cality found in forum text. Despite their small sizes, we also employ these treebanks in adapting the parsers to our target text by using them as supplementary training data to the widely used newswire treebanks for English and French. The following section lists the specific research questions addressed in this set of experiments.

### 1.3.1 Research Questions

The aim of building Foreebank is to understand the characteristics of the Norton forum text from various perspectives. It is also used to measure the amount of noise involved in the parses of this text and to reduce it. Specifically, the following questions are posed:

- *How noisy is the user-generated content of the Norton forum text?*
- *To what extent do user errors in the forum text affect its parse quality?*
- *How noisy is out-of-domain parsing of the Norton forum text?*
- *How effectively can we adapt our parsers to the Norton forum text, both intrinsically and in terms of the accuracy of semantic-based QE which uses semantic role labels from the new syntactic parse trees?*

In the next section, we summarise our approach to answering these questions along with our answers.

### 1.3.2 Summary and Findings

In building Foreebank, we adopt the Penn Treebank (Marcus et al., 1993), the French Treebank (Abeillé et al., 2003), and the English Web Treebank (Mott et al., 2012) annotation guidelines and extend them in a way that accounts for the characteristics of user-generated forum text such as language errors and text style. Concretely, we mark user errors on the parse tree nodes. The adapted annotation strategy enables us to extract useful information from the treebank such as the user error rate in this type of text. It also means that the effect of user errors on the parsing can be examined by comparing the performance of the original and edited versions of

the Foreebank data. The results indicate that user errors account for only a small fraction of the data. However, correcting them can lead to a considerable increase in parsing performance. We additionally conduct a set of experiments to adapt the parsers to Norton forum text using these treebanks, by simply using them as a supplement to the original training data. This method proves to be fairly effective. Finally, based on the idea that more accurate parses will result in more accurate semantic role labelling, we rebuild the semantic-based QE systems using the output of the adapted parsers. Although there is a slight improvement in the performance, it seems that a bigger improvement in parsing is required to be effective in the downstream semantic-based QE task.

## 1.4 Thesis Structure

This thesis comprises nine chapters including the current introductory chapter. The experiments carried out in this thesis are described in Chapters 2 to 8 and the conclusions and suggestions for future work are presented in Chapter 9. Each of the seven main chapters include an introduction followed by a review of the literature related to the problems addressed in the chapter. This is followed by a description of the experiments, a discussion of the results and a summary of the main findings.[2]

Chapter 2 presents the experiments on the use of syntax in machine translation. It describes the data and the experimental setting as well as the automatic and manual comparison of the phrase-based and syntax-based methods. In Chapter 3, we turn our attention to quality estimation of machine translation. This chapter describes in detail the task of quality estimation of machine translations. It then introduces and analyses the data created for the QE experiments throughout the thesis. Chapter 4 presents the experiments on using syntax in quality estimation and their results. These experiments are carried out on two data sets described in Chapter 3. Chapter 5 elaborates on the role of syntax in QE from various perspec-

[2]For easy comparison, all the results of the quality estimation experiments in this thesis are unified in Tables A.1 and A.2 in Appendix A.

tives, including the impact of parser accuracy and the parts played by the source and target syntax. In Chapter 6, we report the experiments on semantic role labelling of French conducted in order to find an optimum SRL solution given the limited available resources. In Chapter 7, we present our experiments on using semantics in quality estimation and their results. This chapter also offers an analysis of the results carried out to discover the reasons why semantic-based QE sometimes fails. Finally, Chapter 8 introduces the Foreebank and its annotation strategy as well as presenting various analyses conducted using this treebank to understand the challenges associated with parsing this type of text. In addition, it describes the evaluation of the parsing performance of the Norton forum text and the experiments on adapting the parsers to this text using the Foreebank. At the end of this chapter, the output of the adapted parsers are used to obtain a new semantic role labelling for the data and rebuild new semantic-based QE systems with the new labelling.

## 1.5 Publications

The majority of the work reported in this thesis has been published at NLP/MT conferences including AMTA 2012, IJCNLP 2013, COLING 2014, *SEM 2014 and SSST 2014. The experiments on syntax-based SMT presented in Chapter 2 are described in Kaljahi et al. (2012). Kaljahi et al. (2014c) describes the syntax-based quality estimation presented in Chapter 4 and the experiments on the role of source and target syntax in Chapter 5. The impact of parser accuracy on the syntax-based quality estimation described in Chapter 5 is published in Kaljahi et al. (2013). The French semantic role labelling experiments are described in Kaljahi et al. (2014a). Finally, the semantic-based QE experiments presented in Chapter 7 are published in Kaljahi et al. (2014b). The SymForum data set has also been made publicly available.[3] We also plan to publish the Foreebank annotation and the parser adaptation experiments.

[3] `http://nclt.dcu.ie/mt/confidentmt.html`

# Chapter 2

# Syntax in Statistical Machine Translation

There has been a long tradition of using syntactic knowledge in statistical machine translation (SMT) (Wu and Wong, 1998; Yamada and Knight, 2001). After the emergence of phrase-based statistical machine translation (Koehn et al., 2003; Och and Ney, 2004), several attempts have been made to further augment these techniques with information about the structure of the language. The motivation behind incorporating syntactic analysis in the translation process is to capture the structural correspondence existing between languages mainly manifesting itself as word order. Examples of such differences include *subject-verb-object* (SVO) versus *subject-object-verb* (SOV), for instance between English and Japanese and long-distance word order such as auxiliary verb translation between English and German.

Phrase-based translation models map continuous phrases, i.e. sequences of strings, in the source languages to continuous phrases in the target language. In transition from phrase-based models to syntax-based models, *hierarchical phrase-based* modelling (Chiang, 2007) supports gaps inside the phrases based on the recursive structure of language but does not concern itself with the linguistic details. On the other hand, syntax-based modelling uses syntactic structures such as parse tree fragments in mapping from source to the target (Galley et al., 2004; Zollmann and Venugopal,

2006). Syntactic information is incorporated into the model from parse trees on the source side (tree-to-string), target side (string-to-tree), or both (tree-to-tree). Other approaches employ dependency treelets (Xiong et al., 2007) or use syntax as a soft constraint in the translation process (Chiang, 2010; Zhang et al., 2011).

Using such linguistic generalisation, however, has proven to be a more complicated task than one might first imagine it to be. While relative improvements over phrase-based baselines have been reported for some language pairs, those baselines seem to remain the best option for other language pairs (DeNeefe et al., 2007; Zollmann et al., 2008). The performance of syntax-based models is affected by errors introduced by existing imperfect syntactic parsers (Quirk and Corston-Oliver, 2006). Moreover, some non-syntactic phrases (e.g. *I'm*) identified by the phrase-based models bring useful information to the translation which are missed by syntax-based models trained on trees obtained using supervised parsing (Bod, 2007). Phrasal coherence between the source and target languages (Fox, 2002) is another factor affecting the performance of syntax-based models. Nevertheless, these models should in theory be better able to capture long-distance reordering — a problem for phrase-based models.

A framework combining such varying techniques can exploit the advantages of all of them while compensating for the weaknesses of each individual method. To accomplish this goal, a more detailed insight into the characteristics of each method may be useful. Towards this objective, we look for possible systematic differences between variants of phrase-based and syntax-based systems via various analysis approaches. More specifically, we compare the output of these systems to discover 1) whether they generate sufficiently different translations for the same sentence in order for their combination to be useful, and 2) whether the syntax-based approaches better handle grammar-related phenomena in translation.

In the rest of this chapter, we first review the work done in the area of syntax-based machine translation in Section 2.1. In Section 2.2, we introduce the data we use for the experiments followed by a presentation of the SMT systems we built in

Section 2.3. In Section 2.4, we compare these systems using automatic evaluation metrics in various ways. In Section 2.5, the systems are compared manually in terms of a number of grammatical and lexical phenomena. Finally in Section 2.6, we offer an error analysis in which the output of different systems for some examples are analysed.

## 2.1 Related Work

Yamada and Knight (2001) argue that string-to-string IBM translation models are only suitable for structurally similar language pairs. They propose a syntax-based model for English-to-Japanese translation, a language pair with different word orders, which uses the conventional noisy channel model for SMT but with syntactic parse trees as its input (tree-to-string model). Working at the constituency node level, the channel takes each node and stochastically reorders all its children, inserts extra words at each node and translates leaf nodes (words). The reordering operation models the word order difference between the two languages and the insertion operation models the structural difference between them, specifically in terms of case marking. The translation operation performs the actual translation on word-by-word basis, ignoring the context.

Carreras and Collins (2009) propose a syntax-based translation approach based on tree adjoining grammar or TAG (Joshi-Schabes-1997), which works by mapping sequences of source strings to parse tree fragments in the target language and allows the integration of a syntactic language model (Charniak, 2001). These fragments are then merged using TAG parsing operations to form a full parse tree resulting in the full translation of the source segment. This method uses discriminative dependency parsing operations while combining the target parse tree fragments to allow a flexible reordering. Their experiments on German-English translation show statistically significant improvements over a phrase-based system when evaluated both automatically using BLEU and manually.

DeNeefe et al. (2007) compare a phrase-based model with another string-to-tree translation approach, in which *translation rules* as the basic units of translation are extracted from a parallel corpora, the target side of which is parsed into phrase structure trees. A translation rule contains sequences of words in its left hand side and syntactic tree fragments in the right hand side, as opposed to the phrase translation pairs in phrase-based translation where both sides contain sequences of words. While the syntax-based model performs better than the phrase-based model on Chinese-English translation, it is shown to be worse on Arabic-English translation. They find that non-lexical rules form only a small fraction of the translation rule table in syntax-based modelling. The string-to-tree modelling in this work is based on their approach.

Zollmann et al. (2008) observe that the gain achieved by hierarchical and syntax-based models can be largely compensated for by increasing the reordering limit in the phrase-based model. They argue that the phrase-based systems over which improvements are reported by Marcu et al. (2006); Chiang (2007); DeNeefe et al. (2007) are restricted to a distortion limit of 4 or 7 words, while their hierarchical or syntax-based systems are able to perform a reordering of 10 words or more. In other words, by allowing those phrase-based systems to move the translated words farther than their position in the source side, they can perform as well as the syntactically-enhanced systems. They also find that, for language pairs involving substantial reordering like Chinese-English, syntactic tree-based models perform better than phrase-based models. However, for relatively monotonic pairs like Arabic-English, all models produce similar results. This is in line with the results reported by DeNeefe et al. (2007).

Experimenting with French-English, German-English and English-German, Auli et al. (2009) compare a phrase-based model to a hierarchical phrase-based model by exploring as much of the search space of both types of models as is computationally feasible. Given that the search spaces are very similar, they conclude that the differences between the two types of models can be explained by the way they score

hypotheses rather than by the hypotheses they produce.

Using the same framework as in this work, Hoang et al. (2009) compare phrase-based, hierarchical phrase-based and string-to-tree models for English-to-German translation. While the phrase-based and hierarchical phrase-based models achieve similar results, they both perform slightly better than the syntax-based model.

Neubig and Duh (2014) suggest that the performance of syntax-based machine translation depends on a number of peripheral factors including the accuracy of syntactic parses, word alignments and the search algorithm. Using a tree-to-string system, they experiment with English to Japanese and Japanese to English translation and find that using the output of a more accurate parser increases the performance of their system. A relatively bigger improvement is gained when the parse trees are replaced with forests, similar to Mi et al. (2008). These improvements lead to a 2 and 1 higher BLEU points for the English to Japanese and Japanese to English translation respectively. Additionally, they observe that more accurate word alignments improve the performance of the syntax-based system, while they do not affect the phrase-based or hierarchical translations. In terms of the search algorithm, they compare *hypergraph* (Heafield et al., 2013) and *cube pruning* (Huang and Chiang, 2007) algorithms and find that the former is more useful for syntax-based translation than the latter which is the most standard search algorithm used for tree-to-string translation. They conclude that syntactic information can be beneficial to machine translation provided that these peripheral factors are considered.

There have been several efforts to exploit the difference between phrase-based and syntax-based models in MT system combination or multi-engine machine translation (MEMT) (Huang and Papineni, 2007). The task, however, has been shown to be difficult. Zwarts and Dras (2008) try to identify what type of sentence can be better translated by a syntax-based model compared to a phrase-based model. Using a classification approach, they separately test three sets of features. Sentence length and system-internal features including decoder output score do not lead to an accurate classifier. They then hypothesise that noisy parse trees may impede the

performance of the syntax-based system and build another classifier based on source sentence length, parser confidence score, and linked fragment count. However, they do not find any correlation between these features and the performance of different systems. Based on their observation that most of the problems in the output are related to reordering, they assume that the syntactic quality of the output could be discriminative in system selection. They port the parse quality features used on the source side to the target side, but again find no improvement.

We build upon previous work by analysing a more comprehensive set of SMT methods. While the majority of the other works reviewed here experiment with only one syntax-based method, we use three different approaches as well as two baseline phrase-based methods. In addition, we perform various comparisons using both automatic and manual judgements to study the difference between these translation methods from different perspectives, rather than simply using automatic evaluation metrics to find the best-performing method. We additionally compare these methods on a diverse set of evaluation data in various automatic and manual ways.

## 2.2 Data

The training data for the translation models of our machine translation systems consist of English-French (En-Fr) and English-German (En-De) Symantec translation memories. These translation memories contain a mixture of Symantec content from product manuals, software strings, marketing materials, knowledge bases and websites. The En-Fr parallel data contains 975,102 sentence pairs and the En-De 1,029,741 pairs with no exact duplicates.

For training language models of both English-German and English-French systems, we use a combination of the target side of their respective translation model training data and a limited amount of user forum text available for each language: 42K sentences for French and 67K sentences for German.

We have two evaluation sets for each language pair:

1. **French translation memory**: 5,000 held-out sentences from the Symantec En-Fr translation memory, split into development (2000) and test (3000) sets
2. **German translation memory**: 5,000 held-out sentences from the Symantec En-De translation memory, split into development (2000) and test (3000) sets
3. **French forum data**: 1,500 sentences taken from the Symantec English online forums, split into development (600) and test (900) sets. These were automatically translated into French using an online translation tool and then post-edited by human translators.
4. **German forum data**: 1,500 sentences taken from the Symantec English online forums, split into development (600) and test (900) sets. Similar to the French ones, these were automatically translated into German using an online translation tool and then post-edited by human translators.

The translation memory data can be considered a superset of forum data in terms of subject matter. However, in terms of style, the forum data is more informal and, given that it is user-generated content, we assume that it exhibits a higher level of ungrammaticality. Because of this difference, we call the evaluation sets taken from the translation memories *in-domain* and those from forum text *out-of-domain*. While the English sides of the in-domain sets are different for each of the language pairs, those of the out-of-domain sets are the same for both pairs.

## 2.3 SMT Systems

We train five statistical machine translation systems, one phrase-based, one hierarchical phrase-based and three syntax-based, as follows:

1. `PB`: a standard phrase-based system (Och and Ney, 2004)
2. `HP`: a hierarchical phrase-based system (Chiang, 2007)
3. `TS`: a tree-to-string syntax-based system (Huang et al., 2006).
4. `ST`: a string-to-tree syntax-based system (DeNeefe et al., 2007).

5. `TT`: a tree-to-tree syntax-based system

We choose these five systems because they are the most widely used methods and can be built using the open source Moses toolkit (Hoang et al., 2009).

The `PB` system was trained using the `grow-diag-final-and` alignment heuristic and used the `msd-bidirectional-fe` reordering model. All other parameters were default including a maximum phrase length of 7 and a decoder distortion limit of 6 when applied. The `HP` system was trained using the default settings. A maximum decoder chart span of 20 was used for the`HP`, `TS`, `ST` and `TT` systems.

To produce syntactic parses needed by the syntax-based systems (TS, ST, TT), we use the `Lorg` parsing system[1] to parse the English and French sides of the corpora. The German side was parsed by the Berkeley parser (Petrov et al., 2006). The Lorg parser is very similar to the Berkeley parser, the main difference being its unknown word handling mechanism (Attia et al., 2010).[2] This parser learns a latent-variable probabilistic context-free grammar (PCFG-LA) from a treebank in an iterative process. PCFG-LAs have been shown to perform well for several languages and domains (Petrov, 2009; Huang and Harper, 2009; Le Roux et al., 2012) and is a state-of-the-art parsing methodology.

We use the Wall Street Journal section of the *Penn Treebank* (Marcus et al., 1994) for training the English parsing model, the *French Treebank* (FTB) (Abeillé et al., 2003) for training the French model and the *Tiger treebank* (Brants et al., 2002) for training the German parsing model.

During the extraction of translation rules, limiting the phrase boundaries to only syntactic constituents imposes a strict constraint on the rule extraction leading to a small rule table. Subsequently, the performance of the syntax-based systems is substantially lower compared to the phrase-based ones. To relax this constraint, we use the SAMT-2 parse relaxation method (Zollmann et al., 2008) implemented in

[1] `https://github.com/CNGLdlab/LORG-Release`

[2] The two parsers achieve Parseval labelled F-scores in the 89-90 range on Section 23 of the Wall Street Journal section of the Penn Treebank. Due to some character encoding issue, the Lorg parser could not be used to parse the German data and this is why the Berkeley parser is used instead.

Moses. In this method, any pairs of adjacent nodes in the parse tree are combined together to form new nodes. This significantly increases the number of extracted rules and consequently the translation accuracy.

All five systems are tuned using minimum error rate training (MERT) (Och, 2003) on the respective developments sets.

## 2.4 Automatic System Comparison

In this section, we compare our SMT systems built in the previous section from various perspectives using automatic evaluation metrics. We first compare their performance using multiple MT evaluation metrics. We then compare the output of each pair of these systems to verify if different systems produce different translations for the same source sentence. This is done at both system and sentence level. Moreover, we mine the n-best output of the systems to search for such differences. We finally build an oracle combined systems based on automatic sentence-level comparison using both 1-best and n-best translations.

### 2.4.1 Multiple Metrics

In order to carry out a reliable comparison, we evaluate the baseline systems at the document level using four widely used metrics: BLEU (Papineni et al., 2002), TER (Snover et al., 2006), GTM (Turian et al., 2003) and METEOR (Denkowski and Lavie, 2011)[3]. The results of the evaluation with these metrics for in-domain and out-of-domain development sets are presented in Table 2.1 and Table 2.2 respectively. The upper half of the tables shows the results of English-French systems and the lower half the results of English-German systems. We report scores on the development sets since our analysis has been performed on these. We used paired bootstrap resampling with 1000 iterations and p-value = 0.01 for all and only BLEU significance tests.[4]

[3]We used all default parameters for evaluation tools. For GTM, we used 1.2 as the exponent.
[4]The tool used is available at `http://www.ark.cs.cmu.edu/MT/`

Table 2.1: Evaluation scores on in-domain development set (translation memory)

| | *En-Fr* | | | |
|---|---|---|---|---|
| | BLEU | TER | GTM | METEOR |
| PB | 0.6140 | 0.3584 | 0.6357 | 0.7436 |
| HP | **0.6188** | **0.3535** | **0.6400** | **0.7457** |
| TS | 0.5919 | 0.3719 | 0.6194 | 0.7284 |
| ST | 0.6013 | 0.3631 | 0.6258 | 0.7334 |
| TT | 0.5783 | 0.3842 | 0.6096 | 0.7168 |
| | *En-De* | | | |
| | BLEU | TER | GTM | METEOR |
| PB | 0.5099 | 0.4911 | 0.5441 | 0.6264 |
| HP | **0.5289** | **0.4676** | **0.5592** | **0.6408** |
| TS | 0.4939 | 0.4923 | 0.5349 | 0.6146 |
| ST | 0.5086 | 0.4753 | 0.5479 | 0.6265 |
| TT | 0.4784 | 0.5059 | 0.5219 | 0.6051 |

Performance is considerably higher for the in-domain evaluation sets compared to the out-of-domain ones and for the En-Fr compared to En-De. Neither of these results are surprising since it is well known that out-of-domain translation is challenging and that English-German translation is more difficult than English-French translation. It is worth noting that the gap between En-Fr and En-De scores on out-of-domain data is bigger than on in-domain data, showing that out-of-domain En-De is a more difficult machine translation setting compared to the others.

The hierarchical phrase-based system (HP) performs better than the others on the in-domain data according to all metrics. This is statistically significant in the case of BLEU scores with p-value $< 0.01$. The gap is more pronounced on the En-De pair, which is an intuitively appealing result because the hierarchical phrase-based model is in theory better able to model the systematic word order differences between English and German than the phrase-based model. The phrase-based system (PB) is the second best performing system on in-domain data.

The string-to-tree system (ST) is the best of the syntax-based systems on in-domain data according to all metrics. The tree-to-tree model (TT), on the other hand, is the worst-performing of these systems, despite its relatively larger transla-

Table 2.2: Evaluation scores on out-of-domain development set (forum text)

| | *En-Fr* | | | |
|---|---|---|---|---|
| | BLEU | TER | GTM | METEOR |
| PB | **0.3044** | 0.6024 | 0.3972 | 0.5335 |
| HP | 0.3032 | **0.5904** | **0.4008** | **0.5341** |
| TS | 0.2907 | 0.6118 | 0.3924 | 0.5202 |
| ST | 0.2982 | 0.6057 | 0.3952 | 0.5248 |
| TT | 0.2900 | 0.6121 | 0.3910 | 0.5166 |
| | *En-De* | | | |
| | BLEU | TER | GTM | METEOR |
| PB | **0.1681** | 0.7428 | 0.3062 | **0.4057** |
| HP | 0.1662 | 0.7384 | 0.3082 | 0.4028 |
| TS | 0.1643 | **0.7197** | **0.3128** | 0.3977 |
| ST | 0.1654 | 0.7286 | 0.3117 | 0.3976 |
| TT | 0.1633 | 0.7358 | 0.3090 | 0.3966 |

tion rule table size. In the case of BLEU scores, these differences are also statistically significant. This shows that less useful rules are extracted by this model compared to the other two models.

On the out-of-domain data, however, the behaviour of the systems is not consistent, with different metrics favouring different systems for different language pairs. On En-Fr, HP is still the best overall, and ST is the best performing syntax-based system (all statistically significant in the case of BLEU). On the other hand, more inconsistent behaviour is observed on En-De: TS scores the best of all according to two (half) of the metrics (though marginally), and HP is no longer the best. However, the BLEU differences are not statistically significant.

### 2.4.2 One-to-one Comparison

Given the same training material, we are interested in the extent to which the methodological differences between these systems lead to different outputs. The more similar the outputs of different systems, the less effective the complex methods (tree-based methods here) will be, compared to the phrase-based method which is usually a baseline in machine translation research. Also, if systems tend to generate

Table 2.3: One-to-one BLEU Scores on in-domain development set (translation memory data)

| | *En-Fr* | | | | *En-De* | | | |
|---|---|---|---|---|---|---|---|---|
| | PB | HP | TS | ST | PB | HP | TS | ST |
| HP | **0.8535** | - | - | - | **0.7576** | - | - | - |
| TS | 0.7799 | 0.7958 | - | - | 0.6817 | 0.7071 | - | - |
| ST | 0.7769 | 0.7917 | 0.7980 | - | 0.6624 | 0.6940 | 0.6778 | - |
| TT | **0.7339** | 0.7430 | 0.8065 | 0.7893 | **0.6405** | 0.6484 | 0.7113 | 0.7068 |

Table 2.4: One-to-one BLEU Scores on out-of-domain development set (forum text)

| | *En-Fr* | | | | *En-De* | | | |
|---|---|---|---|---|---|---|---|---|
| | PB | HP | TS | ST | PB | HP | TS | ST |
| HP | **0.7501** | - | - | - | **0.6207** | - | - | - |
| TS | 0.6640 | 0.7028 | - | - | 0.6023 | 0.6162 | - | - |
| ST | 0.6618 | 0.6959 | 0.6731 | - | 0.5700 | 0.5802 | 0.6191 | - |
| TT | **0.6165** | 0.6344 | 0.7122 | 0.6764 | **0.5211** | 0.5365 | 0.6027 | 0.6014 |

highly similar outputs, their combination cannot yield a noticeably better result.

To inspect this phenomenon for systems built here, we score each system against all others using the BLEU metric. In other words, each system output plays the role of reference translation for the other four systems.

Table 2.3 and Table 2.4 show the comparison results for in-domain and out-of-domain evaluation sets respectively. According to the results, HP and PB are consistently the most similar to each other (highest BLEU), whereas TT and PB are the most different (lowest BLEU). This shows that the difference/similarity between the translation methods reflects the difference/similarity between their output. It cannot be said which of the two syntax-based systems are the most distant ones from each other as it differs according to the data sets. However, TT is usually one side of the pair. In addition, systems produce more divergent output on out-of-domain data and on the En-De pair than on in-domain data and on the En-Fr pair respectively. Considering that the out-of-domain translation as well as the En-De translation is more difficult than their counterparts, it can be concluded that the more difficult

the translation task, the more different output is generated by different translation systems. Based on the observation in the previous section that the performance gap between systems were smaller on the more difficult tasks (e.g. out-of-domain translation), this may indicate that different systems handle different sentences more differently when the translation is more difficult. It should be noted that all systems are built upon the same word alignment, under the same framework (Moses), and make use of the same training data for translation and language models. This can be an important contributing factor to the similarity of the output of these systems.

### 2.4.3 Sentence Level Comparison

To gain further insight into the differences between systems, we compare their output sentence-by-sentence using the TER evaluation metric (Snover et al., 2006). Table 2.5 shows the results of this comparison. The first row displays the number of sentences on which all systems scored the same. The second row contains the number of sentences for which all systems generated exactly the same output sentence. The following five rows, one for each system, present the number of sentence translations on which that system scores the highest (first column), possibly along with other systems, and the number of sentence translations on which that system scores the highest alone (second column). We call the former *any-wins* and the latter *solo-wins*. For example, the phrase-based system (`PB`) ranks first 582 times (any-wins) in total on the in-domain En-Fr evaluation set, out of which it is the only system at the highest rank 130 times (solo-wins).

The any-win ranking is not consistent with the solo-win ranking, especially on the in-domain sets. For example, on in-domain En-De, while `HP` ranks the highest in terms of any-wins (612 sentences), `ST` is the one with the most solo-wins (174 sentences). This may suggest that `HP` is mostly the best on the sentences on which the other systems perform similarly, whereas `ST` is better capable of translating those sentences which are troublesome for the other systems.

In addition, it can be observed that, on about one third of the in-domain sets,

Table 2.5: Sentence-level TER-based System Comparison (On 2000 in-domain and 600 out-of-domain development set samples)

| | *In-domain* | | | |
|---|---|---|---|---|
| | *En-Fr* | | *En-De* | |
| Score ties | 740 (37%) | | 627 (37%) | |
| Exact matches | 738 (37%) | | 578 (29%) | |
| PB Any/Solo wins | 582 (29%) | **130** (7%) | 513 (26%) | 123 (6%) |
| HP Any/Solo wins | **586** (29%) | 95 (5%) | **612** (31%) | 125 (6%) |
| TS Any/Solo wins | 489 (24%) | 103 (5%) | 514 (26%) | 116 (6%) |
| ST Any/Solo wins | 517 (26%) | 125 (6%) | 572 (29%) | **174** (9%) |
| TT Any/Solo wins | 394 (20%) | 94 (5%) | 447 (22%) | 100 (5%) |
| | *Out-of-domain* | | | |
| | *En-Fr* | | *En-De* | |
| Score ties | 32 (5%) | | 35 (6%) | |
| Exact matches | 26 (4%) | | 16 (3%) | |
| PB Any/Solo wins | 190 (32%) | 71 (12%) | 163 (27%) | 51 (9%) |
| HP Any/Solo wins | **244** (41%) | **88** (15%) | 172 (29%) | 43 (7%) |
| TS Any/Solo wins | 173 (29%) | 56 (9%) | **208** (35%) | 73 (12%) |
| ST Any/Solo wins | 177 (30%) | 60 (10%) | 205 (34%) | **78** (13%) |
| TT Any/Solo wins | 160 (27%) | 62 (10%) | 196 (33%) | **78** (13%) |

systems achieve the same scores (score ties), most of them being exactly the same translations (exact matches). The ratio is, however, far less for out-of-domain data sets: only about 4%. Given the performance gap between these two domains (Table 2.1 and Table 2.2), this discrepancy is expected to some degree: the closer the outputs to the reference, the less divergent they can be. However, this large ratio disparity does not seem to be only justified by this fact, suggesting that the real difference between systems is revealed on more difficult tasks. This is in par with the conclusion made in the precious section.

### 2.4.4 N-best Comparison

So far our analysis has been carried out on the best translation returned by each system. We now compare the 500-best (distinct) output of systems. For each evaluation set, Table 2.6 shows the degree of overlap between the n-best outputs of the five systems, in terms of the number and percentage of sentences having a

Table 2.6: 500-best overlaps: number and percentage of sentences having a common translation in their 500-best list as well as the average number of common 500-best translations per sentence across data sets (For 2000 in-domain and 600 out-of-domain development set samples)

| | *In-domain* | | *Out-of-domain* | |
|---|---|---|---|---|
| | *En-Fr* | *En-De* | *En-Fr* | *En-De* |
| Number of sentences | 1579 | 1367 | 202 | 169 |
| Percentage of sentences | 78% | 68% | 33% | 28% |
| Average number of overlaps | 17 | 17 | 4 | 5 |

common translation in their 500-best list as well as the average number of common 500-best translations per sentence (overlaps) across data sets. The figures show that there is larger overlap between the n-bests of the in-domain data than the out-of-domain data and the En-Fr pair than the En-De one. This is consistent with our other observations and appears to suggest that the more difficult the sentences are to translate, the more differently the systems perform on them. The small number of common n-best translations in average shows that the systems generate fairly different n-best output. This suggests that the combination framework can further benefit from the n-best lists.

### 2.4.5 Oracle Combination

Using the sentence-level TER scores for each data set, we select the best translation for each sentence and form the oracle combined output of all systems. In case of score ties, we choose the output of systems in this order: `PB`, `HP`, `ST`, `TS`, and `TT`. The list is sorted by the computational cost of training and translating with each system. We also build an oracle by merging and reranking 500-best translations of all systems using TER scores. The oracle combination outputs are evaluated using all the metrics. The scores are presented in the last two rows of Table 2.7 and Table 2.8. `Oracle 1-best` is the combination of the top translations selected by the systems. The performances of the individual systems are also repeated from Table 2.1 and Table 2.2 in the table (in grey) for comparison.

Table 2.7: Baseline and oracle system combination scores on in-domain development set (translation memory)

| | *En-Fr* | | | |
|---|---|---|---|---|
| | BLEU | TER | GTM | METEOR |
| PB | 0.6140 | 0.3584 | 0.6357 | 0.7436 |
| HP | **0.6188** | **0.3535** | **0.6400** | **0.7457** |
| TS | 0.5919 | 0.3719 | 0.6194 | 0.7284 |
| ST | 0.6013 | 0.3631 | 0.6258 | 0.7334 |
| TT | 0.5783 | 0.3842 | 0.6096 | 0.7168 |
| Oracle 1-best | 0.6658 | 0.2917 | 0.6840 | 0.7818 |
| Oracle 500-best | 0.7770 | 0.1779 | 0.7852 | 0.8616 |
| | *En-De* | | | |
| | BLEU | TER | GTM | METEOR |
| PB | 0.5099 | 0.4911 | 0.5441 | 0.6264 |
| HP | **0.5289** | **0.4676** | **0.5592** | **0.6408** |
| TS | 0.4939 | 0.4923 | 0.5349 | 0.6146 |
| ST | 0.5086 | 0.4753 | 0.5479 | 0.6265 |
| TT | 0.4784 | 0.5059 | 0.5219 | 0.6051 |
| Oracle 1-best | 0.5739 | 0.3858 | 0.6111 | 0.6775 |
| Oracle 500-best | 0.6870 | 0.2584 | 0.7145 | 0.7712 |

As expected, there are large gaps between the best performing systems on each data set and the oracle combinations. The gaps are specially bigger for 500-best lists, indicating that the translation systems have ranked higher quality translations lower than the one they have selected as the best translation. This is consistent with the observation in the previous section, where a considerable difference was found in the 500-best outputs of the systems. In the case of BLEU and for the 1-best combination, the relative improvements are 7% and 9% on in-domain En-Fr and En-De and 10% and 15% on out-of-domain En-Fr and En-De respectively. For 500-best combination, these figures are 16%, 19%, 17% and 27%. Apparently, the benefit from combination increases as the level of translation difficulty increases. This is a further confirmation that the different systems built here behave more differently on more difficult data.

Table 2.8: Baseline and oracle system combination scores on out-of-domain development set (forum text)

| | *En-Fr* | | | |
|---|---|---|---|---|
| | BLEU | TER | GTM | METEOR |
| PB | **0.3044** | 0.6024 | 0.3972 | 0.5335 |
| HP | 0.3032 | **0.5904** | **0.4008** | **0.5341** |
| TS | 0.2907 | 0.6118 | 0.3924 | 0.5202 |
| ST | 0.2982 | 0.6057 | 0.3952 | 0.5248 |
| TT | 0.2900 | 0.6121 | 0.3910 | 0.5166 |
| Oracle 1-best | 0.3343 | 0.5408 | 0.4265 | 0.5585 |
| Oracle 500-best | 0.3921 | 0.4717 | 0.4687 | 0.6117 |
| | *En-De* | | | |
| | BLEU | TER | GTM | METEOR |
| PB | **0.1681** | 0.7428 | 0.3062 | **0.4057** |
| HP | 0.1662 | 0.7384 | 0.3082 | 0.4028 |
| TS | 0.1643 | **0.7197** | **0.3128** | 0.3977 |
| ST | 0.1654 | 0.7286 | 0.3117 | 0.3976 |
| TT | 0.1633 | 0.7358 | 0.3090 | 0.3966 |
| Oracle 1-best | 0.1935 | 0.6700 | 0.3376 | 0.4248 |
| Oracle 500-best | 0.2457 | 0.6049 | 0.3791 | 0.4750 |

## 2.5 Manual System Comparison

In the previous section, we compared systems based on scores generated using automatic metrics. It is interesting and useful to know how these different systems handle various linguistic phenomena in translation. For example, one common argument in comparing syntax-based and phrase-based systems is that the former generates more fluent word order in the output. In order to investigate these assumptions, we select 100 sentences from each development set and compare the outputs of two of the systems, namely HP and ST, for each of these sentences. 50 of the selected sentences are the solo-win cases of HP and the other 50 are those of ST (see section 2.4.3). The reason why these two systems are selected is that HP is the overall best performing system, and ST is the best performing syntax-based systems according to various evaluations in the previous section.

Each data set was evaluated by a linguist using eight error categories, adapted from those used by Dugast et al. (2007) to evaluate post-editing changes. The eval-

uators were asked to count the number of errors in each output sentence under each category. While they were given the reference translation, they were not constrained to it and were allowed to compare against the closest correct translation to the output itself. We believe that this can better reflect the real performance of the systems, as it is not limited to a single reference, though we might lose some correlation with automatic metrics. The following are the categories used in the evaluation, the first half of which can be considered to be grammar-related and the second half lexical.

1. *Verb tense:* wrong verb tense translations
2. *Gender/number agreement:* wrong gender and number agreements
3. *Local word order:* wrong local word orders
4. *Long-distance word order:* wrong long-distance word orders
5. *Mis-translated:* wrong word/phrase translations including wrong sense and unusual usage
6. *Untranslated:* words/phrases transferred to the output without translation
7. *Spurious translation:* words/phrases added to the output without any counterpart in the source
8. *Missing translation:* words/phrases in source ignored by the system

The results of the manual evaluation are shown in Table 2.9. We observe the following:

- Since verb tense and gender/number agreement are handled in a methodologically similar way, the two categories can be collapsed for the purposes of comparison. From this point of view, `HP` generates better output. This gap is more pronounced on the in-domain data.
- French word order (both local and long distance) is better handled by `ST` and German word order by `HP`.
- Though no generalizable pattern is seen for mis-translation, it can roughly be said that `ST` is less erroneous than `HP` on this category.

Table 2.9: Manual evaluation results: number of errors by each system on each data set; the lower number of errors are marked in boldface for each category/setting.

| | *In-domain* | | | | *out-of-domain* | | | |
|---|---|---|---|---|---|---|---|---|
| | *En-Fr* | | *En-De* | | *En-Fr* | | *En-De* | |
| | HP | ST | HP | ST | HP | ST | HP | ST |
| Verb tense | 13 | **11** | **1** | 3 | 25 | 25 | 21 | **20** |
| Gender/number agreement | **27** | 34 | **25** | 31 | **64** | 66 | **60** | 63 |
| Local word order | 29 | **25** | **24** | 31 | 53 | **47** | **93** | 99 |
| Long-distance word order | 7 | **6** | **3** | 4 | 8 | **5** | **70** | 82 |
| Mis-translated | 85 | **84** | 61 | **55** | 185 | **177** | 207 | 207 |
| Untranslated | 10 | **9** | 15 | **13** | **92** | 95 | 81 | **72** |
| Spurious translation | **21** | 26 | 29 | **22** | **16** | 21 | **25** | 35 |
| Missing translation | **35** | 41 | 42 | **31** | 18 | **13** | 140 | **122** |
| *Sum* | **227** | 236 | 201 | **194** | 461 | **449** | **697** | 700 |

- `ST` outputs overall fewer untranslated words. However, the gap is marginal. It, on the other hand, tends to generate more spurious translations. The only exemption is on in-domain En-De. On the other hand, `HP` misses more words and phrases.

It appears that no confident conclusion can be made based on the above observations. However, contrary to what one might expect, the syntax-based model is not necessarily better than the hierarchical model in treating syntactic phenomena in translation. The next section provides a closer scrutiny of the internal behaviour of the systems.

## 2.6 Error Analysis

In order to discover the differences between the phrase-based and syntax-based translation metheods, we look at the translation rule tables of each system and follow their decoding process. It can be observed that relaxation blurs the boundaries between the phrase-based and syntax-based models. The rules in the syntax-based models are also based on ad-hoc phrases and the only difference is in the set of non-terminals.

Table 2.10: Example of verb tense translation by two systems

| | |
|---|---|
| Source | If you choose to continue, **you will need to** set the options manually from the Altiris eXpress Deployment Server Configuration control panel applet. |
| Reference | Si vous décidez de continuer, **vous devrez** configurer les options manuellement à partir de l'applet du panneau de configuration Altiris eXpress Deployment Server. |
| HP output | Si vous décidez de continuer, **vous devrez** configurer les options manuellement à partir de l'applet Altiris eXpress Deployment Server Configuration Control Panel. |
| HP rule application | X -> *will need to* $X_1$ *from* \| *devrez* $X_1$ *à partir de* |
| ST output | Si vous décidez de continuer, **vous devez** configurer les options manuellement dans l'applet de panneau de configuration de Altiris eXpress Deployment Server. |
| ST rule application | X -> , *you will need* VPINF \| SENT\PP -> , *vous devez* VPINF |

Table 2.10 illustrates an example in which neither of the rules used by the systems to translate *you will need* is built upon a syntactic phrase. Nevertheless, unlike ST, HP translates it correctly. It is worth noting that there were eight similar rules in the rule table of ST (including the one used in the example) covering the span *, you will need X*, half of which could translate it correctly. However, due to a higher score, this rule was selected.

Another example concerning output word order, which is a major motivation behind incorporating syntax in machine translation, is presented in Table 2.11. Although the spans on which the ST rules have been applied are syntactic in this case, the first two rules have been wrongly chosen resulting in an invalid output word order. On the other hand, HP has correctly parsed the input and applied appropriate rules, leading to a correct output word order.

Despite the pitfalls of the relaxation method used here, the syntax-based models which are built using original parse trees suffer from limited translation rule coverage and produce significantly lower results. This suggests that the syntax-based approaches implemented in Moses are not sufficient to fully leverage the syntactic

Table 2.11: Example of output word order of two systems

| | |
|---|---|
| Source | blocking adult websites |
| Reference | blocage des sites web pour adultes |
| HP output | Blocage des sites Web réservés aux adultes |
| HP rule application | X -> *adult* X \| X *réservés aux adultes*<br>S -> S X \| SX<br>S -> <s> \| <s><br>X -> *blocking* \| *blocage des*<br>X -> *websites* \| *sites web* |
| ST output | Adulte de blocage de sites Web |
| ST rule application | X -> NP//NP *websites* \| NP//ADJ -> NP//NP *sites web*<br>X -> *blocking* NC \| NP//NP -> NC *de blocage de*<br>X -> *adult* \| NC -> *adulte* |

information in translations and other approaches of using syntax in a less restrictive manner are required. For example, Zhang et al. (2011) use syntax as a soft constraint by augmenting the source side of a string-to-tree model with SAMT-style syntactic labels, instead of first parsing the source side and then relaxing the annotation, and a fuzzy rule matching algorithm instead of requiring the source sentence syntax to match the extracted rules. Alternatively, forest-based translation (Mi et al., 2008) can loosen the constraints by providing a large set of parse tree options to the translation rule extractor which can in turn lead to a bigger rule table. Additionally, while the single tree translation is prone to parsing errors, the availability of a large set of parse trees in forest-based translation can help account for the parsing noise during decoding.

## 2.7 Summary and Conclusion

We built a number of SMT systems using phrase-based, hierarchical phrase-based and syntax-based methods. We compared these systems both automatically and manually looking for a systematic difference between their output which could help

understand how these different methods work and could be used to exploit the benefit of each method in a combined framework.

The results of various automatic evaluations showed that hierarchical phrase-based models are overall slightly better than the others. One-to-one and sentence-by-sentence comparison and oracle combination of the output of all models showed that the more difficult the translation problem, the more different their output and the greater the gain to be achieved by combining outputs.

Manual analysis of the outputs and translation process showed that there was no obvious systematic difference between syntax-based and non-syntax-based modelling, mostly due to the relaxation of syntactic constraints on translation rule extraction. This makes it difficult to find features to be utilized in combining these models, despite the potential gain which was observed in their oracle combination. One way to handle this problem is to use quality estimation to choose the best from among the outputs of all systems for each source segment, provided that reliable estimations exist. To this end, we turn our attention to the quality estimation of machine translation in the next chapter.

# Chapter 3

# Quality Estimation of Machine Translation

*Quality Estimation* (QE) of machine translation is the task of measuring the correctness of a MT system output without any reference translation. The absence of a reference translation is the point at which QE diverges from *machine translation evaluation*, a more established task in the field of machine translation. A growing amount of research has recently been carried out on QE for MT, with approaches differing with respect to the nature of the quality scores being estimated, the learning algorithms used or the features chosen to represent the translation pairs in the learning framework. The WMT shared tasks on quality estimation (Callison-Burch et al., 2012; Bojar et al., 2013, 2014) has especially boosted this research by enabling to compare the performance of several different approaches to QE in a unified evaluation framework.

The crucial aspect of quality estimation is the metric by which the quality is measured, i.e. the definition of quality. Much of the previous work in QE for MT has focused on learning to predict human evaluation scores as a measure of quality. Various levels of score granularity have been used, ranging from simply good/bad or correct/incorrect translation (Blatz et al., 2004) to more fine-grained 5-grade scores (LDC, 2002; Callison-Burch et al., 2012). Such elaborated scores have been

used to target different aspects of quality such as the amount of effort required for post-editing the translation by a human or its fluency and adequacy. Fluency captures how well the translation can be read, and adequacy captures how much of the meaning is retained during the translation (Pierce and Carroll, 1966). In addition to such discrete scores, continuous scores, such as the time needed by a human to post-edit (Allen, 2003) the translation or even automatic MT evaluation metric scores (Bojar et al., 2013), have been used to express the translation quality. Moreover, there have been works which compare the output of various translation systems for the same input and estimate the quality ranking, rather than explicitly assigning them a score (Bojar et al., 2013).

The translation quality can be judged at various output levels. One can evaluate every word in the output as being correct or incorrect (Ueffing et al., 2003) or alternatively as requiring a post-editing action such as a deletion or substitution (Bojar et al., 2013). One can also assign the quality score to the output itself instead (Blatz et al., 2004). The highest level of granularity is the document which can be useful for large-scale commercial applications (Soricut and Echihabi, 2010).

The features used in quality estimation can be categorised into a) source-based, b) target-based, c) source-to-target-based and d) MT-system-based. Each of these features aim at capturing translation quality from different perspectives. While source-based features (e.g. sentence length) generally capture the difficulty of the source text for translation, target features mainly target the translation quality directly. However, when used together, they can capture the correspondence between the source and target similar to source-to-target features. MT system features, on the other hand, are used to encode the internal process of the translation and also indicate the confidence of the MT system in producing the translation.

In the rest of the chapter, we first discuss the related work in quality estimation of machine translation. We then describe the two data sets we build to use in quality estimation experiments throughout the rest of the thesis starting from the next chapter.

## 3.1 Related Work

The early work on quality estimation of machine translation can be attributed to Gandrabur and Foster (2003). In a translation prediction task, they estimate a confidence score derived from the conditional probability of the correctness of n-grams (up to 4-grams) in the translation of a sentence. They use various features for capturing the difficulty of the source sentence 1) in general, 2) for translation and 3) for translation using a specific model. These features include n-gram language perplexity and probability, number of possible translations per word, number of translation hypotheses for the word (by the current model).

Further research targeting the quality estimation at word level was carried out by Ueffing et al. (2003). They built a system which tagged a word in the translation as correct or incorrect with the aim of guiding the post-editing process or interactive translation. This system uses the information extracted from the word graph (Ueffing et al., 2002) and the n-best list of MT output to compute the word posterior probabilities as quality scores.

Blatz et al. (2004) extend the quality measurement level from word to sentence. They perform a binary classification of translated sentences into *correct* and *incorrect*. The correctness is defined based on a threshold set upon two different automatic MT evaluation metrics, namely *WERg* and *NIST*. The data set they use is comprised of about 6500 Chinese sentences paired with each of their 1000-best English translation hypotheses output by a phrase-based machine translation system. The features they use are extracted from the MT model itself, the n-best list output of the model and the source and target sentences. The n-best list features include the rank of the translation hypothesis in the list, the ratio of its score to the best score in the list as well as the average hypothesis length. The source sentence features include its length and n-gram frequency statistics, and the target sentence features include language model scores and word frequencies. They additionally use features based on the source/target correspondence such as IBM Model 1 probabilities.

Since the WMT 2012 workshop[1], a shared task in quality estimation has been organized each year. In the 2012 shared task (Callison-Burch et al., 2012), there were two subtasks, one of which was estimating the required post-editing effort on a 5-point scale, and the other ranking the translations in the test set based on their quality. The task was performed on a data set of 1832 and 422 training and test sentences respectively from news text translated from English to Spanish using a Moses model. The post-editing effort for each translation was the average of three scores assigned by three different human evaluators based on the amount of post-editing actions such as deletion, insertion or substitution, required to correct the translation. The shared task organizers provided a set of 17 features to be used as the baseline. These features mainly concern the surface characteristics of the source and translation and include highly discriminative features such as source sentence length. Table 3.1 lists the features. This feature set built a strong QE system, with only a few submitted systems able to improve over it. These features are simple and shallow in terms of the linguistic information they carry. The best-performing system (Soricut et al., 2012) used system-dependent features extracted from decoder logs, POS tags and pseudo-reference features, and performed an extensive automatic feature selection.

The following shared task in WMT 2013 (Bojar et al., 2013) changed 5-point scores to *human-targeted* TER (HTER) scores (Snover et al., 2006), where instead of human evaluators assigning a score from 1 to 5, they minimally post-edited the machine translations. These post-edits were finally used as references against which the original translations were scored using the TER metric. It also introduced new tasks: estimating real-valued post-editing time with a new data set, ranking outputs of several MT systems for a single source sentence and estimating the translation quality at the word level. The word level QE included two settings. In one setting, each word in the translation was identified as correct or incorrect. In the other one, the post-editing action required to correct the word, including *keep as is*, *delete*

[1] `http://www.statmt.org/wmt12/`

Table 3.1: WMT 2012 17 baseline features

| | **Constituency** |
|---|---|
| 1 | Number of tokens in the source sentence |
| 2 | Number of tokens in the target sentence |
| 3 | Average source token length |
| 4 | Language model probability of the source sentence |
| 5 | Language model probability of the target sentence |
| 6 | Type/token ratio: average number of occurrences of the target word within the target hypothesis (averaged for all words in the hypothesis) |
| 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 (as given by IBM 1 table thresholded so that Prob($t\|s > 0.01$) weighted by the inverse frequency of each word in the source corpus |
| 9 | Percentage of unigrams in quartile 1 of frequency (lower frequency words) in a corpus of the source language |
| 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 a corpus of the source language |
| 14 | Percentage of trigrams in quartile 4 of frequency of source words a corpus of the source language |
| 15 | Percentage of unigrams in the source sentence seen in a corpus of the source language |
| 16 | Number of punctuation marks in source sentence |
| 17 | Number of punctuation marks in target sentence |

and *substitute* was predicted. In addition to English-Spanish, this workshop added German-English to the MT system ranking subtask.

Compared to 2012, more systems outperformed the baseline. Some of the best-performing systems in different tasks used some kind of syntactic and semantic features. Many systems performed feature selection to find the (most) effective features from among mostly large numbers of features. One of the findings of this shared task was that the best systems performed competitively to reference-based evaluation in ranking the MT systems.

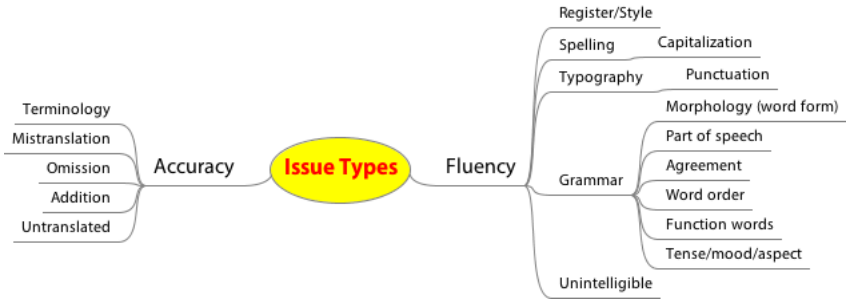



Figure 3.1: Fine-grained word translation error categories of WMT 2014 QE shared task (adapted from Bojar et al. (2014))

In the latest of these shared tasks (Bojar et al., 2014), the subtask of predicting post-editing effort returned, but with only 3 classes of such effort, one of which identifies the translation as near miss which means it can be fixed by post-editing. Three new language pairs were introduced for this subtask: Spanish-English, English-German and German-English. However, the provided data sets were small in size: the source sides of the training and test data sets comprised only 350 and 150 sentences respectively for the new language pairs though there were 3 translations per source sentence. The training and test data sets for the existing language pair, English-Spanish, had 954 and 150 sentences respectively with 4 translations per source sentence.

Another change compared to the previous shared task involves the word-level QE task. In this task, there were three different but related quality estimation settings differing in their level of granularity. The first setting was a binary classification of translated words to good or bad translation, similar to the previous shared task. The second setting included a 3-class classification: 1) good, 2) *accuracy* error and 3) *fluency* error. In the third setting, each of the accuracy and fluency error classes were broken down into more fine-grained class as shown in Figure 3.1. These error types are adapted from *MQM* (Multidimensional Quality Metrics)[2].

Fewer systems participated in this shared task than in 2013. The English-German and German-English language pairs seem to have been easier than the English-Spanish and Spanish-English ones in the post-editing effort prediction task,

[2]`http://www.qt21.eu/launchpad/content/training`

as more systems could beat the baseline. In the binary classification setting of the word-level task, only for English-German could a system outperform a baseline which assigned *bad*, the most common class, to all translated words. In the 3-class setting in this task, no system could beat a baseline assigning the most common class for the English-Spanish and German-English language pairs. On the other hand, in the MQM error classification setting, the baseline ranked the highest only for the German-English language pair. The results show that the more fine-grained the error classes are the better the systems performe compared to the baseline, probably due to an uneven coarse-grained class distribution in the data.

Various approaches were taken by different systems to learn the estimations. Many systems used QuEst (Specia et al., 2013)[3], a system which can be used to extract a wide variety of QE features. Most of the submitted systems used some kind of syntactic features most of which were extracted from POS tags, such as those based on POS n-gram language models. The system which ranked highest in the majority of tasks and settings (Bicici and Way, 2014) also reports using parse tree structures obtained by CCL (Common Cover Links) among a couple of million features. It is worth mentioning that, in estimating the post-editing efforts none of the systems took into account features related to the post-editors such as familiarity of the post-editor with the domain, post-editing experience, etc. (De Almeida and O'Brien, 2010), as such information were not made available by the shared task.

In this work, we address the quality estimation at the sentence level. We use a variety of quality measures including automatic and human-targeted MT evaluation metrics and 5-grade adequacy and fluency scores. Our focus in terms of the information used for estimation is on the linguistic aspects of the source and target sides of the translation, namely syntax and semantics. Syntax and semantics have been previously used in building quality estimation systems for machine translation by Quirk (2004), Hardmeier et al. (2012), Avramidis (2012), and Pighin and Màrquez (2011) among others. We will elaborate on these works in Chapter 4 and 7.

[3] `http://www.quest.dcs.shef.ac.uk/`

## 3.2 Data

The goal of this work is to address the utility of syntax and semantics in estimating the quality of machine translation of user-generated content (UGC). The user-generated content comes from the Norton English user forum and the translation direction is from English to French. To build a quality estimation system appropriate for this text domain, style and language pair, a data set with similar characteristics and containing human evaluation of its machine translation quality (or alternatively human post-edits) is required. We build such a data set as it does not exist. We select the data from monolingual Norton user forums. The selected segments are machine-translated and the quality scores are obtained using both scoring against their human post-edits as in the WMT 2013 QE shared task, and human evaluators judging the adequacy and fluency of the translations. This data set is called *SymForum* and described in Section 3.2.1.

The information we use as clues of machine translation quality in this work is derived from syntactic and semantic analyses of the source and target text. Such analyses for user-generated text and machine translation output is prone to noise. One pitfall of using the noisy information extracted from erroneous text is that the conclusions drawn on its results can consecutively be inaccurate. On the other hand, this information can be reliably extracted for well-formed text using state-of-the-art syntactic parsers (Collins, 1999; Klein and Manning, 2003; Charniak and Johnson, 2005; Petrov et al., 2006), which achieve above 90 $F_1$ points for English. Therefore, it is reasonable to first apply syntax-based QE to well-formed text and then move to UGC data, even though the challenge of dealing with machine translation output still exists for both cases.

Statistical data-driven parsers are the state of the art in syntactic parsing. Such parsers are mainly tailored to the edited newswire text due to the availability of annotated resources for this domain, and their performance deteriorates when applied to other domains (McClosky et al., 2010) and especially unedited text (Foster, 2010).

Unfortunately, not enough human-evaluated machine-translated data exists for QE in the English-to-French translation direction in this domain. The only available English-to-French data set which contains human judgements and can be used in quality estimation are as follows:

- CESTA (Hamon et al., 2007), which is selected from the Official Journal of the European Commission and also from the health domain. In addition to the domain (and style) difference to newswire (the domain on which our parsers are trained), a major stumbling block which prevents us from using this data set is its small size: only 1135 segments have been evaluated manually.
- WMT 2007 (Callison-Burch et al., 2007), which contains only 302 distinct source segments (each with approx. 5 translations) only half of which is in the news domain.
- FAUST[4], which is out-of-domain and difficult to apply to our setting as the evaluations and post-edits are user feedbacks, often in the form of phrases/ fragments.

An alternative formulation is to use automatic evaluation metrics instead of human evaluation as the measure of quality to be estimated. Although automatic MT evaluation metrics have been criticised for their low correlation with human scores, they have the advantage of being easier to obtain thanks to existence of many parallel corpora in a variety of domains. In addition, it has been shown that human judgements are not necessarily consistent (Snover et al., 2006). This replacement enables us to easily build a data set on the same domain in which our syntactic parsers are built. We choose the News development data set released for the WMT 2013 translation task (Bojar et al., 2013)[5]. We call this data set the *News* data set and describe it in Section 3.2.2.

[4] `http://www.faust-fp7.eu/faust/Main/DataReleases`
[5] `http://www.statmt.org/wmt13`

Table 3.2: Characteristics of the source and machine-translated side of SymForum data set

| | English | French |
|---|---|---|
| Average sentence length | 14.2 | 15.9 |
| Sentence length SD | 9.8 | 10.8 |
| Type/token ratio | 16.9% | 16.9% |

### 3.2.1 The SymForum Data Set

We randomly select 4500 segments from a large collection of monolingual English Norton forum text containing 3 million segments[6]. Each segment is produced by segmenting the content of a forum thread into single sentences using the statistical model included in the NLTK[7] toolkit. Since sentence boundary is naturally ambiguous, especially in the case of user-generated content extracted from HTML sources, some segments do not represent full sentences; they may be truncated or merged. We finally tokenize the segments using our own rule-based tokeniser which is built and tuned for Norton forum text and Symantec translation memories. In order to be independent of any one translation system, we translate the data set with the following three systems and randomly choose 1500 distinct segments from each.

- ACCEPT[8]: a phrase-based Moses system trained on Symantec translation memory supplemented with the WMT 2012 releases of Europarl and News Commentary corpora.
- SYSTRAN: a proprietary rule-based system (Enterprise Server 6.8.0 with Symantec domain dictionaries)
- Bing[9]: an online translation system (used on 24 February, 2014)

[6]We choose this amount because 1) it is a reasonable size for training and testing a reliable machine learning model and 2) it is affordable in terms of both computational time and human annotation labour.

[7]urlhttp://www.nltk.org/

[8]`http://www.accept.unige.ch/Products/D_4_1_Baseline_MT_systems.pdf`

[9]`http://www.bing.com/translator` on 24 February, 2013

Table 3.2 shows some statistics extracted from the source and target of the data set. We measure the translations quality in two ways described here.[10]

#### 3.2.1.1 Human-targeted Scores

One method to evaluate the translations is to minimally post-edit them by humans so that an adequate and fluent translation is achieved. The MT output is then scored against these post-edits as references using an automatic metric. These scores are known as *human-targeted* scores and shown to correlate better with average human evaluation scores than one human score does with another (Snover et al., 2006). Furthermore, the same study shows that human-targeted metrics also correlate well with reference-based metrics. This is especially a useful feature for us as, in the next chapter, we will import the systems built using reference-based metrics for the News data to this setting.

Therefore, the output of machine translation systems are post-edited by one human post-editor. As the source text is unedited and prone to ungrammaticality (and even incomprehensibility itself) and also to limit translator subjectivity, we emphasize *good enough* quality instead of perfect quality to keep MT assessment as realistic as possible. We define good enough as *comprehensible* (i.e. the main content of the message be understood), *accurate* (i.e. it communicates the same meaning as the source text), but without being stylistically compelling. The text may sound like it was generated by a computer, syntax might be somewhat unusual, grammar may not be perfect but the message is accurate. In other words, the *minimum* edits should be done to achieve good enough quality.

To this end, we ask post-editors to comply with the following guidelines during post-editing:[11]

[10]The data set is publicly available at `http://www.computing.dcu.ie/mt/confidentmt.html`

[11]The post-editing guidelines are based on the TAUS/CNGL guidelines for achieving "good enough" quality downloaded from `https://evaluation.taus.net/images/stories/guidelines/taus-cngl-machine-translation-postediting-guidelines.pdf`. Post-editing is done by a professional translator who is a native French speaker. However, they do not necessarily possess the domain knowledge of Norton products.

Table 3.3: Human-targeted evaluation scores for SymForum data set at the document level, segment level average and standard deviation (SD)

| | 1-HTER | HBLEU | HMeteor |
|---|---|---|---|
| Document level | 0.6907 | 0.5577 | 0.7241 |
| Segment-level Average | 0.6976 | 0.5517 | 0.7221 |
| Segment-level SD | 0.2446 | 0.2927 | 0.2129 |

- Aim for semantically correct translation.
- Ensure that no information has been accidentally added or omitted.
- Use as much of the raw MT output as possible.
- Translate appropriately for obvious source misspelling.
- No need to implement corrections that are of a stylistic nature only.
- No need to correct punctuation if it reflects the source punctuation.

We then score each sentence translation against its post-edited version at segment level using BLEU[12], TER[13] and Meteor[14] and call them HBLEU, HTER and HMeteor (human-targeted scores[15]). Note that Since TER scores change in the opposite direction (i.e. the lower the better), we present 1-HTER to be better comparable to the BLEU and Meteor. In addition, there is no upper bound for TER scores unlike the other two metrics. Scores higher than 1 occur when the number of errors is higher than the segment length. To avoid this, scores higher than 1 are cut-off to 1 before being converted to 1-HTER. The document level scores as well as average scores for the entire data set together with their standard deviations are presented in Table 3.3. According to the scores, the highest standard deviation belongs to HBLEU scores. Meteor scores, on the other hand, have the lowest standard deviation.

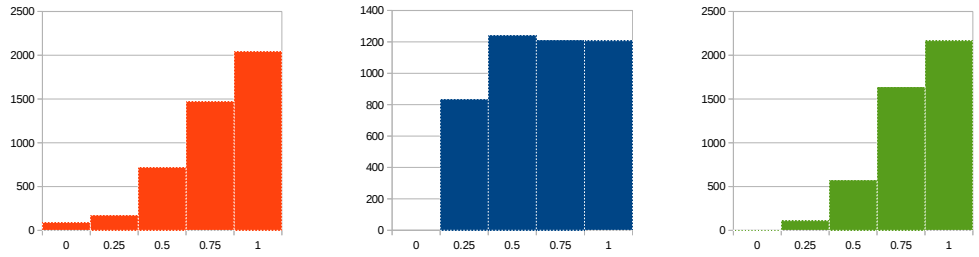
We draw the score distribution histograms for these three metrics in Figure 3.2. HTER and HMeteor scores are distributed similarly which is very different from the

[12]Version 13a of MTEval script was used at segment level.

[13]TER COMpute 0.7.25: `http://www.cs.umd.edu/~snover/tercom/`

[14]Meteor 1.4: `http://www.cs.cmu.edu/~alavie/METEOR/`

[15]It should be noted that the original notion of human-targeted scores introduced by Snover et al. (2006) assumes that the post-editor is presented with a reference translation, which is not the case in this work.




(a) HTER (b) HBLEU (c) HMeteor

Figure 3.2: Human-targeted score distribution histograms of the SymForum data

Table 3.4: Adequacy and fluency score interpretation

| | Adequacy | Fluency |
|---|---|---|
| 5 | All meaning | Flawless Language |
| 4 | Most of meaning | Good Language |
| 3 | Much of meaning | Non-native Language |
| 2 | Little meaning | Disfluent Language |
| 1 | None of meaning | Incomprehensible |

HBLEU score distribution which tends to be even across score bins. More than half of the HMeteor scores are in the highest bin suggesting it as a lenient metric.

#### 3.2.1.2 Adequacy and Fluency Scores

An alternative way of obtaining human judgements of the quality of a MT output is to ask human evaluators to assign it a score indicating a quality measure such as required post-editing cost or adequacy/fluency. We choose the second measure here as it is closer to the purpose of our study which is the use of syntax and semantics in estimating the quality. Adequacy measures how much the meaning of the source is delivered in the MT output and fluency measures how fluent the translation is.

We asked three professional human translators, who were native French speakers, to assess the quality of MT output in terms of adequacy and fluency on a 5-grade scale. This scoring scheme is adapted from LDC (2002) and the interpretation of the scores is given in Table 3.4. Each evaluator was given the entire data set for evaluation. We therefore collected three set of scores and averaged them to obtain the final scores for each MT output sentence.

Similar to human-targeted scores, we plot the score distribution histograms of

Table 3.5: Manual evaluation scores for SymForum data set (segment level average and standard deviation (SD))

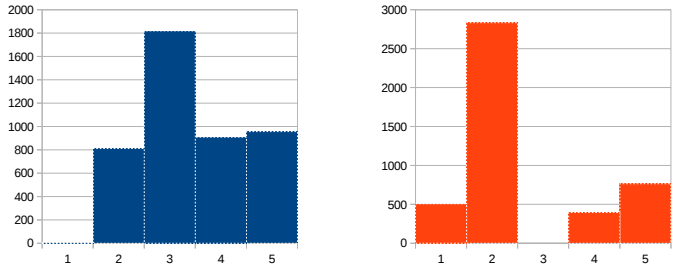
| | Adequacy | Fluency |
|---|---|---|
| Segment-level Average | 0.6230 | 0.4096 |
| Segment-level SD | 0.2488 | 0.2780 |

each metric for each of the annotators and their average in Figure 3.3. The figures show that adequacy scores tend to be higher than the fluency ones according to all evaluators. While the first evaluator assigns the mean adequacy score of 3 to the majority of translations, the other two use the highest score of 5 more than all others. Consistently, translations not conveying any of the source sentence meaning, i.e. adequacy score 1, compose the smallest fraction of the scores. This is reflected in the average adequacy scores (Figure 3.3d.1), where there is only 1 score less than or equal to 1.

In terms of fluency, the first and third evaluators find the majority of the translations as good as score 2 while the second one assigns the mean score of 3 for most of them. Interestingly, the first evaluator does not give the fluency score 3 to any translation. Additionally, while there are more fluency scores of 1 in sum than 4 and 5, when they are averaged, only a small number of score 1 remain. This suggest that the evaluators largely disagree on incomprehensibility of the translations. The consensus between the annotators is further verified later in this section by calculating the inter-annotator agreement.

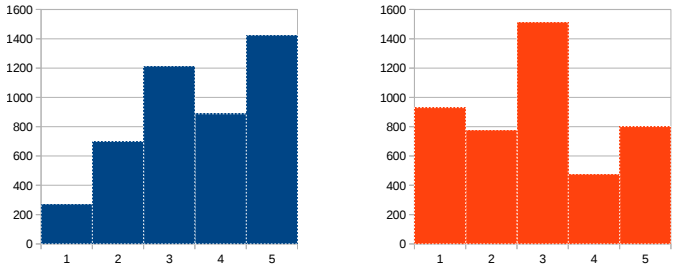
In terms of average scores (Figure 3.3d), adequacy scores are more evenly distributed compared to fluency scores. Most of the translations are of above average adequacy. However, their fluency is mostly below the average.

In order to be easily comparable to human-targeted scores, we scale these scores to the [0,1] range, i.e. adequacy/fluency scores of 1 and 5 are mapped to 0 and 1 respectively and all the scores in between are accordingly scaled. The averages of these scores for the entire data set together with their standard deviations are presented in Table 3.5.



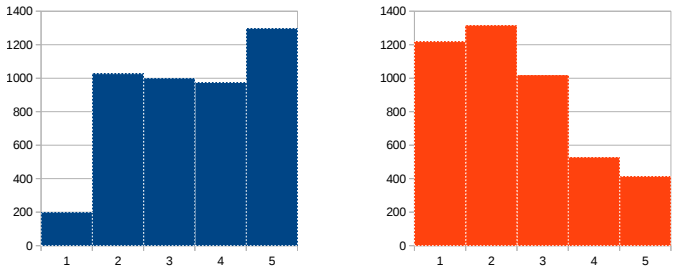

(a.1) Adequacy (a.2) Fluency

(a) `Evaluator 1`



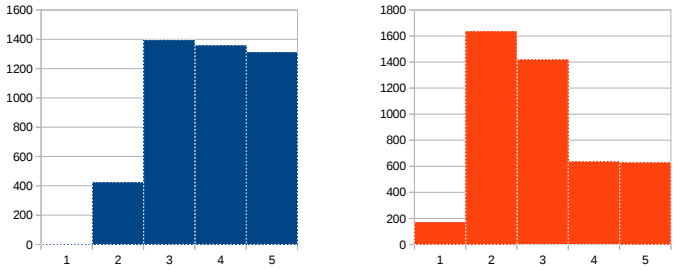

(b.1) Adequacy (b.2) Fluency

(b) `Evaluator 2`




(c.1) Adequacy (c.2) Fluency

(c) `Evaluator 3`




(d.1) Adequacy (d.2) Fluency

(d) `Average scores`

Figure 3.3: Adequacy and fluency score distribution histograms of the SymForum data set, for each evaluator and for their average

Table 3.6: Pearson r between pairs of metrics on the entire 4.5K data set

| | 1-HTER | HBLEU | HMeteor | Adequacy | Fluency |
|---|---|---|---|---|---|
| 1-HTER | - | - | - | - | - |
| HBLEU | 0.9111 | - | - | - | - |
| HMeteor | 0.9207 | 0.9314 | - | - | - |
| Adequacy | 0.6632 | 0.7049 | 0.6843 | - | - |
| Fluency | 0.6447 | 0.7213 | 0.6652 | 0.8824 | - |

The average weighted Kappa inter-annotator agreement for adequacy scores is 0.65 and for fluency scores is 0.63. We use weighted Kappa instead of plain Kappa to account for close evaluation scores. The reason is that the difference between scores of 5 and 4 is not equal to the difference between 5 and 2. While both of these are regarded as the same by plain Kappa, weighted Kappa can account for the closeness of the scores. Specifically, we consider two scores of difference 1 as 75% agreement instead of 100%. All the other differences are considered to be disagreement. Though this still seems to be strict, the weighted Kappa values are in the substantial agreement range.

Once we have both human-targeted and manual evaluation scores together, it is interesting to know how they are correlated. We calculate the Pearson correlation coefficient r between each pair of the five metrics and present them in Table 3.6.

Interestingly, unlike what is generally expected, HBLEU has the highest correlation with both adequacy and fluency scores among human-targeted metrics. HTER on the other hand has the lowest correlation. Moreover, HBLEU is more correlated with fluency than with adequacy which is the opposite to HMeteor. This is expected according the definition of BLEU and Meteor.

A high correlation can be seen among the human-targeted scores and between the manual evaluation scores. Interestingly, the correlations among the former are higher ($> 0.90$). Although the high correlation between the adequacy and fluency could partially be related to both scores being from the same evaluator (albeit for each of the three evaluation rounds), it indicates that if either fluency or adequacy of the MT output is low (or high), the other tends to be low (or high) as well.

Finally, the data set is randomly split into 3000 training, 500 development and 1000 test segments. We use the development set for tuning model parameters and building hand-crafted feature sets, and the test set for testing model performance and analysis purposes.

### 3.2.2 The News Data Set

To be as comparable as possible, the News data set is created in a similar way to the SymForum data set. We randomly select 4500 parallel segments from the WMT13 News development data set. These segments are translated using the same systems described in section 3.2.1 but with different settings as follows:

- ACCEPT: trained on a parallel corpus created using the translations by Translators without Borders[16] and supplemented with the WMT 2012 releases of Europarl and News Commentary corpora.
- SYSTRAN: Enterprise Server 6.8.0 without Symantec domain dictionaries
- Bing: on 11 March, 2013

Similar to the SymForum data set, 1500 distinct segments from the output of each system is randomly selected to build the final set. However, unlike the SymForum data set, these translations are scored against pre-translated references of the target side of the corpus instead of their post-edits. The scoring is done at the segment level using BLEU, TER and Meteor with the same settings as in the SymForum data set. Note that, similar to SymForum data set, TER scores are cut-off at 1 and converted to 1-TER.

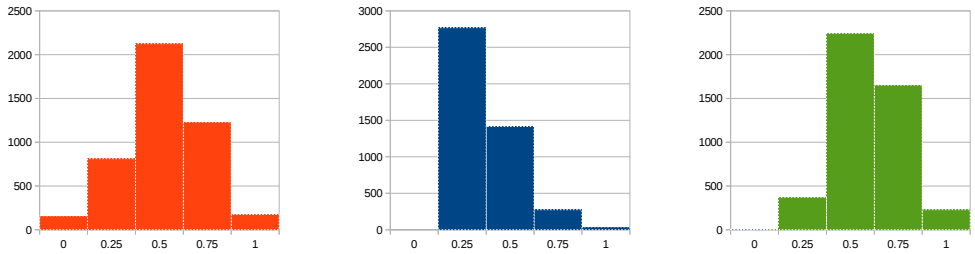
Table 3.7 shows the document level scores and the average and standard deviation of segment level scores. In contrast to the SymForum data set, BLEU has the lowest standard deviation (slightly lower than Meteor) and 1-TER the highest.

We plot the histogram of the score distribution as shown in Figure 3.4. Though not directly comparable, there is a remarkable difference between the distribution of

[16] `http://translatorswithoutborders.org/`

Table 3.7: Automatic evaluation scores for News data set at document level, segment level average and standard deviation (SD)

| | 1-TER | BLEU | Meteor |
|---|---|---|---|
| Document level | 0.4179 | 0.2577 | 0.4779 |
| Segment level average | 0.4087 | 0.2335 | 0.4707 |
| Segment level SD | 0.1951 | 0.1610 | 0.1655 |




(a) 1-TER (b) BLEU (c) Meteor

Figure 3.4: Score distribution histograms of the News data set

scores in these figures and those of the same metrics for the SymForum data set in Figure 3.2. The most obvious difference is that human-targeted metrics had a high tendency to assign high scores to the translations in that data set compared to their automatic variations here. This is expected because the reference translations for that data set are obtained by post-editing and are much closer to the MT output.

The majority of BLEU scores are lower than 0.5 while that is the opposite for Meteor, showing that BLEU is a stricter metric than Meteor. On the other hand, 1-TER scores are distributed more evenly around the average with a slight tendency to fall above the average. This is also reflected in the average scores in Table 3.7. Interestingly, BLEU identifies only 40 translations as perfect. Moreover, BLEU scores have less diversity compared to the other two metrics. Overall, considering also the distributions of scores in the SymForum data set, Meteor seems to be the most lenient metric among all used here.

Similar to the SymForum data set, we extract statistics from the source and translation of the News data set which are presented in Table 3.8. Comparing the figures to those in Table 3.2, it can be seen that the News sentences are longer in average than the SymForum sentences (by around 8 words). Furthermore, the higher

Table 3.8: Characteristics of the source and machine-translated side of News data set

| | English | French |
|---|---|---|
| Average sentence length | 22.2 | 24.2 |
| Sentence length SD | 12.3 | 13.3 |
| Type/token ratio | 21.3% | 20.7% |

type/token ratio for the News data set indicates a higher level of lexical variation in this data set compared to the SymForum.

Finally, analogous to the SymForum data set, the data set is randomly split into 3000 training, 500 development and 1000 test segments.

## 3.3 Summary and Conclusion

We introduced the task of quality estimation for machine translation, QE, which has recently received considerable attention by machine translation researchers. We explored variations of the task differing in terms of translation granularity and quality measure. Most of the work has been done on estimating the translation quality at the sentence level although word level and document level have also been addressed. Various measures have been used to define the translation quality including both automatic and manual, discrete and continuous, and targeting adequacy, fluency, post-editing time, effort and action. We also surveyed some of the research carried out in this area, especially the quality estimation shared tasks of the WMT workshop. A wide variety of features has been used in building quality estimation systems submitted to these shared tasks. Most of the features capture the surface characteristics of the source and target and use language modelling resources. Part-of-speech tags are the most popular syntactic features utilized by such systems.

Quality estimation data sets built upon human evaluation or post-edits are limited to only a few language pairs and domains and are also relatively small in size. For example, there is no reasonably sized data set for our language pairs, text domain

and style of interest which are English-French, technical security software support and user-generated forum text respectively. Due to its importance to the project, we built such a data set. We used three different machine translation systems and two different evaluation strategies, one based on automatically scoring these translations against their human post-edits of these translations (human-targeted scoring) and the other based on evaluating their adequacy and fluency by human evaluators.

The goal of this work is to use the information derived from syntactic and semantic analyses of the source and target. The tools to obtain such analyses are built for edited newswire text, which is out-of-domain and out-of-style for our data set. To circumvent the resulting noise affecting the conclusions made on the usefulness of such information for quality estimation, we built another data set selected from newswire. However, we relied on automatic metrics as quality indicators for this data set instead of human evaluation as they are readily accessible.

We offered several types of analysis of quality scores obtained using these scoring methods for each data set. We calculated the correlation of different metric scores on the SymForum data set and found that the human-targeted scores correlate best with each other (above 90 Pearson r) and that adequacy and fluency correlate with each other (below 90 Pearson r). However, the correlation between these two metric types is lower (mostly in the 60s).

We will use both SymForum and News data sets in Chapter 4 to investigate the use of with syntax in quality estimation and in Chapter 5 where we analyse the syntax-based QE in terms of the effect of parsing accuracy and the role of source and target syntax. We will also use the SymForum in Chapter 7 where we use semantics in quality estimation.

# Chapter 4

# Syntax-based Quality Estimation

It is reasonable to assume that syntactic information is useful in quality estimation of machine translation as a way of capturing the syntactic complexity of the source sentence, the grammaticality of the target translation and the syntactic symmetry between the source sentence and its translation. This assumption has been borne out by previous research which has demonstrated the usefulness of syntactic features for quality estimation of English-Spanish machine translation (Hardmeier et al., 2012; Avramidis, 2012; Rubino et al., 2012). Inspired by this, we design and experiment with quality estimation systems which use syntactic information extracted from both translation source and target.

Syntactic information can be derived from various grammar formalisms each emphasising a different syntactic aspect of language. *Constituency* or *phrase structure* grammar is a well studied syntactic formalism, which recursively parses a sentence into its constituents or phrases, in a tree structure. *Dependency* grammar is another formalism which captures the relations between words in a sentences. These relations can also be represented in a tree structure. We choose to use the constituency and dependency syntax in these experiments as they have shown to be useful in many tasks including quality estimation and there are several resources and tools available for them.

We examine two different methods of encoding syntactic information in quality

estimation as well as their combination: *tree kernels* and *hand-crafted features*. Tree kernels (Collins and Duffy, 2002; Moschitti, 2006) automatically extract the useful information in the syntactic trees and use them in learning the underlying task. Therefore, they eliminate the need for feature engineering which is the common practice in classic machine learning methods. On the other hand, hand-crafted features must be designed and tuned before being used by the learning algorithm. However, they offer more flexibility in deciding what information to be included in or excluded from the learning model, as well as a higher computational efficiency due to a smaller feature space. These two methods can also be combined in a unified framework which has been found useful (Moschitti, 2006).

The experiments are first carried out using the News data set described in the previous chapter, to rule out from conclusion the additional level of noise introduced by parsing Norton forum text which is out-of-domain to the corpora on which the parsers are trained. The same experiments are then replicated with the SymForum data set which is in the target domain of this thesis.

The rest of this chapter is laid out as follows: we first review the previous work using syntax in quality estimation of machine translation in Section 4.1 and in Section 4.2 and Section 4.3 we describe the syntax-based QE experiments with the News and SymForum data sets respectively. Each section includes the experiments using tree kernel, hand-crafted features and their combination, all compared with baseline systems.[1]

## 4.1 Related Work

Features extracted from parser output have been used before in training QE for MT systems. Quirk (2004) uses a single syntax-based feature which indicates whether a full parse for a sentence could be found. The parser generates the logical form

[1]For easy comparison, all the experimental results in this chapter are also presented in Tables A.1 and A.2 in Appendix A, beside other QE results, for the SymForum and News data sets respectively.

representation (LF) of the sentence and is applied to only the source text to capture its difficulty of translation. LF is chosen as the syntactic representation because the MT system evaluated in these experiments is built upon this formalism. The feature is combined with other non-syntax-based features to classify a translation as high or low quality. Various learning algorithms including perceptrons, support vector machines, decision trees and linear regression are used, among which the best performance was achieved by the linear regression system. Although a feature selection is carried out for this algorithm, it is not mentioned whether this syntactic feature is selected in the final feature set.

Hardmeier et al. (2012) employ tree kernels to predict the 1-to-5 post-editing cost of a machine-translated sentence in the WMT 2012 QE shared task setup (Callison-Burch et al., 2012). They use tree kernels derived from syntactic constituency and dependency trees of the source side (English). However, they only use dependency trees of the translation (Spanish) because proper constituency parsing models were not available for this language. They use tree kernels hoping that they help quality estimation by 1) capturing structures in the parse tree of source sentence which are difficult to translate and 2) identifying uncertain constructions in the parse tree of the translated sentence. In order to be used as tree kernels, they convert the dependency trees so that the labels are moved from the edges to separate nodes. The converted trees contain word forms, dependency relations and also POS tags. They use *subset tree kernels* (Collins and Duffy, 2002) for constituency trees and *partial tree kernels* (Moschitti, 2006) for dependency trees.[2] When used alone, the syntactic tree kernels cannot outperform the baseline which consists of the 17 features introduced in Section 3.1 of Chapter 3. When combined with the baseline features, they are able to reduce the prediction error by around 7%. However, adding more features which are mainly adapted from the work by Specia et al. (2009) to this combination does not bring additional information when evaluated on the test set.

[2]Unlike subset tree kernels, partial tree kernels allow the right hand side of a production rule to be split.

They also note that the tree kernels receive a lower weight by the learning algorithm compared to other features. This system however ranked second in the shared task.

Rubino et al. (2012) employ a variety of syntactic features in their QE system for predicting 1-to-5 post-editing cost of the MT output. These features are mainly those previously used by Wagner et al. (2009) to distinguish between grammatical and ungrammatical English sentences and consist of three types of features: 1) POS tag n-gram frequencies in a reference corpus, 2) the output of a rule-based LFG parser (Maxwell and Kaplan, 1996) which uses a hand-crafted broad-coverage grammar of English (Butt et al., 2002), and 3) the output of a statistical constituency parser (Charniak and Johnson, 2005) trained on three different corpora, the Wall Street Journal (WSJ) section of Penn Treebank (Marcus et al., 1993), a version of the WSJ corpus which has been automatically distorted by inserting errors into the sentences and their union. The LFG parser features include whether or not the sentence could be parsed, the number of possible parses and parsing time. The constituency parser features include the difference in the parser log probability between the trees output by the three statistical parsing models and structural differences between the trees measured using various parser evaluation metrics. All three sets of features are extracted from the source side. From the target side, however, only the POS tag features are extracted accompanied by other features calculating the frequency of each specific POS tag in the translated segment, the proportion of target-side words assigned more than one tag and the proportion of those unknown to the tagger. The last set of syntactic features are extracted from the output of LanguageTool[3] which checks the grammar and also the style of the input for errors using a set of predefined rules[4]. These features are used alone and in combination with other types of features in a variety of machine learning settings. Although the syntax-based features cannot outperform the 17 baseline features of the WMT 2012 QE shared task, they achieve the best performance when combined

[3] http://languagetool.org/

[4] http://languagetool.org/languages

together in some of the settings. The syntactic features are also present in all other best performing settings.

Specia and Giménez (2010) argue that quality estimation and reference-based evaluation of machine translation are complementary and attempt to combine them to improve the correlation of the latter with human judgements at segment level. In their QE system, they use POS tag 3-gram language model probabilities extracted from the MT output as features. Note that their system contains reference-based metrics which use features built upon POS tags, syntactic chunks and dependency and constituency structure to build linguistically-informed automatic MT evaluation metrics. These metrics are included in the *Asiya* toolkit (Giménez and Màrquez, 2010). The contribution of the syntactic features to the complete systems is not examined.

Avramidis (2012) builds a series of classification and regression models for estimating post-editing effort using syntactic features in combination with other features. These syntactic features include constituency parse log-likelihood and confidence, parse n-best list size, average confidence of n-best parse trees and frequency of each particular syntactic label. These features are extracted from the source and target text as well as their value ratios. Language quality scores produced by LanguageTool are also used which includes grammar checking of the source and translation segments. All the features undergo a variety of feature selection processes resulting in several features sets. The syntactic features appearing in the two best feature sets they report include a few features from the grammar checking tool plus those extracted from the constituency parses. The constituency parse features include only the frequency of some syntactic labels such as `S`, `CC` and `NN`.

In a similar vein, Gamon et al. (2005) train a classifier to distinguish between human and machine translation. They use the class probability output by this classifier as a quality indicator for the MT output: the more likely it is to be a human translation, the higher quality it is. To build the classifier, they use features extracted from the output of the French *NLPWin* (Heidorn, 2000) system. NLPWin

is a linguistic analysis tool which provides syntactic and semantic analysis of the input and has been used as the natural language processor in Microsoft products such as Word Grammar Checker and natural language query interfaces. The features are extracted by applying the tool on a mixture of machine and human translations and include POS tag trigrams, context free grammar (CFG) production rules and semantically motivated syntactic features such as definiteness. The top 2000 features according to their log likelihood ratio with respect to the class label are chosen. The quality scores (class probability) obtained by this classifier achieve low correlations with fluency and adequacy scores (0.09 and 0.12 respectively) on their test set. They multiply these scores with language model perplexity scores for each sentence and achieve a higher correlations of 0.37 and 0.42 with the fluency and adequacy scores respectively.

Syntax has also been widely used in reference-based evaluation of machine translation. Liu and Gildea (2005) develop a number of syntax-based metrics with the main goal of capturing the fluency of the translations. *STM* (subtree metric) and *TKM* (tree kernel metric) are based on the constituency parse trees of the reference and the hypothesis. The former is inspired by BLEU but counts common subtrees of fixed depth instead of n-grams. TKM extends STM to include all possible subtrees using *convolution kernels* (Collins and Duffy, 2002). They also build *HWCM*, a metric based on headword chains derived from dependency parse trees of the reference and the hypothesis. This metric also works in a similar way to BLEU but uses the headword chain n-grams instead of word n-grams. At the sentence level, all the metrics except TKM outperform BLEU in terms of correlation with 1-to-5 scale human judgements of fluency. HWCM also correlates better than BLEU with the overall quality evaluated by human.

Giménez and Màrquez (2007) argue that the lexical overlap between the reference and hypothesis used by many metrics is neither sufficient nor necessary to judge the quality of translation. Instead, they suggest that the similarity should be measured at a more abstract linguistic level. They define *linguistic elements* as opposed to

lexical elements (word forms) which can capture different kinds of linguistics phenomena at different levels of granularity. Considering syntax as one such linguistic phenomena, they develop and compare several metrics using linguistic elements at the POS tagging and phrase chunking levels, as well as those adapted from other syntax-based metrics including HWCM and STM described above. They find that metrics based on deep syntactic structure such as HWCM- and STM-based ones correlate better than lexical metrics such as BLEU and Meteor with human evaluation scores. However, those based on shallow syntax (e.g. POS tagging and phrase chunking), perform at the same level as or lower than the lexical metrics.

Owczarzak (2008) uses LFG dependencies of the reference and hypothesis to develop another syntax-based evaluation metric. This metric beats all other automatic metrics such as BLEU, TER and Meteor as well as HWCM (Liu and Gildea, 2005) in terms of correlation with human fluency judgements at the segment level but not at the system level. Owczarzak also finds that the quality of the MT output is not correlated with the accuracy of the metric concluding that parsing ill-formed MT output does not negatively influence the correlation of the metric with human scores.

More recently, Yu et al. (2014) propose a metric called *RED* which only requires the dependency parses of the reference segment to avoid the noise associated with the parsing of machine translation output. The metric uses two different dependency structures, one based on the headword chains of Liu and Gildea (2005) and the other based on *fixed-floating* structures (Shen et al., 2010). Headword chains capture the long-distance dependencies and fixed-floating structures represents local continuous n-grams, where *fixed* structures consist of the subtrees under the root node and *floating* consist of contiguous siblings of a common head. Both fixed and floating structures must be complete constituents. To compute the score without having the dependency tree of the translation hypothesis, the metric extracts the word sequences from each headword chain and fixed-floating subtree, and searches the translation for those n-grams. In the case of headword chains, only the order of the

words in the sequence is important, while for the fixed-floating subtrees, they need to be continuous in the translation. They also augment the metric with external resources to account for stem, synonym and paraphrases matches in a similar way to Meteor. RED outperforms BLEU, TER and HWCM at both the system and sentence level translation ranking for most language pairs of their experiments. The augmented metric (REDp) with a parameter tuning can also outperform Meteor at the system level and achieve a comparable performance at the sentence level.

## 4.2 Syntax-based QE with News Data Set

The syntactic structure of a sentence can be represented in various linguistic formalisms which can be roughly categorized as being based on *constituency* or *phrase-structure* grammar, or based on *dependency* grammar. In constituency grammar, the structure of a sentence is recursively constructed from phrases as constituents in the form of a tree, while in dependency grammar, it is expressed as the relations between words. However, these two types of structures are related to each other as the dependency structure of a sentence can be deterministically derived from its constituency structure. In this chapter, we use the syntactic information of the source and its translation encoded in these two formalisms to build a quality estimation system.

The common way to utilise such information in quality estimation is by encoding them as *features* in a machine learning framework. These features capture the particularities of the syntactic structure of the sentence represented by the constituency and dependency trees. A set of features is designed through a feature engineering process which identifies features and determines which features are most useful in predicting the quality of a translation.

Another method is to directly use these trees in a *tree kernel* framework (Collins and Duffy, 2002; Moschitti, 2006). This approach allows exponentially sized feature spaces (e.g. all subtrees of a tree) to be efficiently modelled and has shown

to be effective in many natural language processing tasks including parsing and named entity recognition (Collins and Duffy, 2002), semantic role labelling (Moschitti, 2006), sentiment analysis (Wiegand and Klakow, 2010) and quality estimation of MT (Hardmeier et al., 2012).

Although there can be overlaps between the information captured by these two methods, each of them can capture information that the other one cannot. In addition, while tree kernels involve minimal feature engineering, hand-crafted features offer more flexibility and better computational efficiency (albeit depending on the size of the feature set). Moschitti (2006) shows that combining the two approaches is beneficial. We use both hand-crafted features and tree kernels separately and combined together.

For parsing the data into their constituency structures, we use the `Lorg` parsing system also used in the SMT experiments described in section 2.3 of Chapter 2. We train the English parser on the training section of the Wall Street Journal (WSJ) section of the *Penn Treebank* (PTB) (Marcus et al., 1993). The French parser is trained on the training section of the *French Treebank* (FTB) (Abeillé et al., 2003).

Dependency parses can be obtained by directly parsing the data using a dependency parser (Nivre et al., 2006; McDonald et al., 2006) or, alternatively, by converting from the constituency parses. In the former case, the parser needs to be trained on a dependency treebank. Since a manually built dependency treebank is not available for English and French, it is obtained by automatically converting from a constituency treebank. Therefore, since we already have the constituency parses of our data available, we choose the second method. We convert English constituency parses using the `Stanford` converter (de Marneffe and Manning, 2008) and French parses using `Const2Dep` (Candito et al., 2010).

In the experiments, we evaluate the performance of the QE models based on two measures: Root Mean Square Error (RSME) for how accurate the predictions are and Pearson correlation coefficient (r) for how well the model differentiates

between the quality of different translations.[5] To compute the statistical significance of the performance differences between QE models, we a form of paired bootstrap resampling (Efron and Tibshirani, 1993). In particular, we randomly resample (with replacement) a set of $N$ instances from the predictions of the two systems, where $N$ is the size of the evaluation set. We repeat this sampling $N$ times and count the number of times each of the two settings is better in terms of each measure (RMSE and `Pearson r`). If a setting is better more than 95% of the time, we consider it statistically significant at $p < 0.05$.

In the following sections, we first describe our baseline systems and then the quality estimation systems built using tree kernels, hand-crafted features and a combination of both.

### 4.2.1 Baseline QE Systems

In order to verify the performance of syntax-based QE, we build two baselines. The first baseline (`B-Mean`) uses the mean of the segment level evaluation scores in the training set for all instances. In the second baseline (`B-WMT`), the 17 baseline features of WMT 2012 QE shared task described in Section 3.1 of Chapter 3 are used. `B-WMT` is considered a strong baseline as the system that used only these features was ranked higher than many of the participating systems.

We use support vector regression implemented in the `SVMLight` toolkit[6] to build `B-WMT`. The Radial Basis Function (RBF) kernel is used based on the results of preliminary experiments. The results for both baselines are presented in Table 4.1.

As expected, the `B-WMT17` baseline is statistically significantly better than a mere mean baseline. TER scores seem to be the hardest to predict and Meteor scores the easiest. A notable correlation between the RSME scores of the three metrics and the standard deviations of these metrics presented in Table 3.7 of Chapter 3 can be seen. BLEU scores have the lowest standard deviation in the data set and the

[5]The experiments show that neither of these measures is enough on its own to judge the performance.

[6]http://svmlight.joachims.org/

Table 4.1: Baseline system performances measured by Root Mean Square Error (RSME) and Pearson correlation coefficient (r)

| | BLEU | | 1-TER | | Meteor | |
|---|---|---|---|---|---|---|
| | RSME | r | RSME | r | RSME | r |
| `B-Mean` | 0.1626 | - | 0.1965 | - | 0.1657 | - |
| `B-WMT` | 0.1601 | 0.1766 | 0.1949 | 0.1565 | 0.1625 | 0.2047 |

best RMSE is associated with their prediction. In contrast, TER scores have the highest standard deviation and their prediction achieves the worst RMSE. Overall, we can see a low Pearson correlation for all metrics using the `B-WMT` baseline which indicates the difficulty of the task.

### 4.2.2 Syntax-based QE with Tree Kernels

Tree kernels are kernel functions that compute the similarity between two instances of data represented as trees based on the number of common fragments between them. Therefore, the need for explicitly encoding an instance in terms of manually designed and extracted features is eliminated, while benefiting from a very high-dimensional feature space. Moschitti (2006) introduces an efficient implementation of tree kernels within a support vector machine framework. Instead of extracting all possible tree fragments, the algorithm compares only tree fragments rooted in two similar nodes. This algorithm is made available through `SVMLight-TK` software[7], which is used in this work.

In order to extract tree kernels from dependency trees, the labels on the arcs must be removed. Following Tu et al. (2012), the nodes in the resulting tree representation are word forms and dependency relations.[8] An example is shown in Figure 4.1. A word is a child of its dependency relation to its head. The dependency relation in turn is the child of the head word. This continues until the root of the tree.

[7] `http://disi.unitn.it/moschitti/Tree-Kernel.htm`

[8] They also examined adding POS tags to this structure. However, the new information did not show to be useful. They suggest that this information is already captured by the dependency representation.

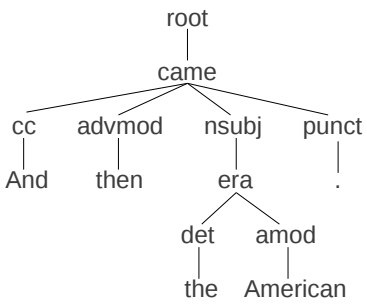



Figure 4.1: Tree Kernel Representation of Dependency Structure for *And then the American era came.*

Based on preliminary experiments on our development set, we use *subset* tree kernels (Moschitti, 2006). Unlike subtree kernels, subset tree kernels also allow tree fragments where the leaves are non-terminal categories. Additionally, we use the sequential summation of the kernels computed between two corresponding trees of two instances. In other words, if two training or test instances contain $n$ trees each, the combined kernel value is computed by as follows:

$$K_T(I_1, I_2) = \sum_{i=1}^{n} k(T_{1i}, T_{2i})$$

where $I_1$ and $I_2$ are the instances, $T_{1i}$ and $T_{2i}$ are the $i_{th}$ trees of $I_1$ and $I_2$ respectively, $k(T_{1i}, T_{2i})$ is the kernel between these two trees and $K_T$ is the combined kernel. We tune the $C$ parameter for Pearson r on the development set. All the other parameters are left default.

We build a system with all four parse trees for every training instance, which includes the constituency and dependency parse trees of the source and target text. In other words, each training (and test) instance contains four parse trees presented sequentially to the learning algorithm. Table 4.2 shows the performance of this system which is named `SyTK`. The `B-WMT` baseline is also presented in this table for comparison (in grey). As can be seen, `SyTK` substantially outperforms the baseline, all the differences being statistically significant.

In order to examine their complementarity, we combine these tree kernels and the baseline features of `B-WMT`. The combined kernel is computed in a similar way

Table 4.2: QE performance with syntactic tree kernels SyTK and their combination with WMT baseline features B+SyTK.

| | BLEU | | 1-TER | | Meteor | |
|---|---|---|---|---|---|---|
| | RSME | r | RSME | r | RSME | r |
| B-WMT | 0.1601 | 0.1766 | 0.1949 | 0.1565 | 0.1625 | 0.2047 |
| SyTK | 0.1581 | 0.2437 | 0.1888 | 0.2774 | 0.1595 | 0.2715 |
| B+SyTK | **0.1570** | **0.2696** | **0.1879** | **0.2939** | **0.1576** | **0.3111** |

explained above for only trees. Specifically, the kernel for the feature vectors is similarly computed and at the end is summed with the tree kernel value computed above. More formally:

$$K_S(I_1, I_2) = K_T(I_1, I_2) + \sum_{i=1}^{m} k_v(V_{1i}, V_{2i})$$

where $K_T$ is the tree kernel as computed above, $V_{1i}$ and $V_{2i}$ are the $i_{th}$ hand-crafted feature vectors of $I_1$ and $I_2$ respectively, and m the size of the feature vector, $k_v(V_{1i}, V_{2i})$ is the kernel between these two vectors, and $K_S$ is the combined kernel.[9] Therefore, the representation of the training and test instances in this setting contain four parse trees followed by a vector of 17 features.

The combined system is named B+SyTK in Table 4.2. All the scores obtained by the combined system are statistically significantly better than when only tree kernels are used, rendering the combination successful.

### 4.2.3 Syntax-based QE with Hand-crafted Features

In this section, we focus on feature engineering for syntax-based quality estimation. We start with nominating *feature templates* that represent the constituency and dependency structure of the sentence from various points of view. Basically, each feature template contains two features, one extracted from the source and the other

[9]In SVMLight, tree kernels and hand-crafted features can be combined by either summation or multiplication of their contributions. We use summation based on our preliminary tuning on the development set. The contribution of the tree kernels and hand-crafted features are given the same weight.

from the target sides of the translation. Features are either numerical, where their values are real numbers, or nominal, where their values are strings. Numerical feature templates can be extended by extracting the relations (e.g. ratio or difference) between the source and target side feature values as features.

Some feature templates are parametric meaning that there can be variations of them by changing the value of the parameters. For example, the non-terminal label (phrase category; e.g. NP, VP, etc.) is a parameter for non-terminal label count (`non-terminal-label-count`) feature template. Therefore, it expands to several *feature sub-types*, one for each non-terminal-label.

As in `B-WMT`, we use support vector machines (SVM) to build the QE systems using these hand-crafted features. SVM requires feature values to be numerical. Consequently, nominal features should be converted to numbers beforehand. For this purpose, we binarize these features by converting each feature value observed in the training set to a binary feature. For instance, the label of the root node of the constituency tree as a feature can get a value from a set of 11 non-terminal labels appearing in the root nodes of the constituency trees of our English News training data. Each of these 11 labels is converted to a feature. If, for example, the root node label of a tree is `SBAR`, the value of feature with the same name will be 1 and that of all the other 10 features will be 0.

For some nominal features, the set of possible values can be large. For example, there are more than 7000 unique POS 3-grams in our English News training data. Therefore, there will be a large number of binarized features created from such features. Not only does this high dimensionality reduce the efficiency of the system, it also affects the performance due to the sparsity of such features, which is exacerbated by the relatively small size of our training data set. To tackle this issue, we impose a frequency cutoff on these features: after binarization, we keep only those which fire for more than a percentage threshold of the training data. This threshold is set empirically for each feature using the development set.[10]

[10]A number of thresholds are tried and at the end the most restrictive ones with at least one

The following lists our syntax-based feature templates. Each feature template has a name followed by its description. There are 14 constituency-based feature templates in total (including POS-based ones) which are presented first, followed by 11 dependency-based features. Parametric feature templates are expanded to several feature subtypes. Unless otherwise specified, numerical feature templates additionally include features capturing the ratio and difference of the source and target features as mentioned above.

`constituency-tree-root:` the label of the root node of the constituency tree

`constituency-tree-height:` the height of the constituency tree which is the number of edges from the root node to the farthest terminal (leaf) node

`constituency-tree-size:` the number of nodes in the constituency tree

`constituency-parse-probability:` the log probability of the constituency parse assigned by the parser

`constituency-pseudo-reference-score:` the Parseval $F_1$ score of the constituency parse tree with respect to a parse tree produced by the Stanford parser (Klein and Manning, 2003). This feature aim at capturing the complexity of the input as well as the grammaticality of the output using parser agreement measured by Parseval $F_1$ score.

`constituency-root-cfg-rule-rhs:` the right hand side of the CFG production rule expanding the root node of the constituency tree.

`constituency-cfg-rules:` all non-lexical and lexical CFG production rules expanding the constituency tree nodes

`average-constituency-tree-arity:` the average arity of the non-lexical CFG production rules expanding the constituency tree nodes (i.e. average number of children of nodes)

`non-terminal-label-count:` the number of times a specific non-terminal label appears in the constituency tree. This feature template expands to subtypes, one for each non-terminal label, which include verb phrases, noun phrases, preposi-

feature left are used.

tional phrases, adjective phrases, adverb phrases, conjunction phrases and sentential phrases. In addition, for English, there are three more features for each of WH, particle, and parenthetical phrases, for which there is no equivalent in the FTB.[11]

`pos-ngrams:` POS n-grams including unigram, 3-gram and 5-gram feature subtypes each expanding to two further subtypes, one for original POS tags and one for universal mappings. The universal mapping introduced by Petrov et al. (2012) maps original POS tags of the PTB and FTB to a set of 12 abstract tags.

`pos-lm-score:` POS n-gram scores against language models[12] trained on the POS tags of the respective treebanks for each language. There are three parameters involved in this feature template expanding it into 12 feature subtypes: n-gram order (3- and 5-grams), LM score type (probability, log probability and perplexity), POS tag set (original and universal).

`universal-pos-count:` the count of each universal POS tag (includes 12 subtypes).[13]

`first-verb-location:` location of the first verb POS tag in the sentence in terms of the token distance from the beginning

`pos-tag-ngram-quartile-frequency:` the average number of POS n-grams in each n-gram frequency quartile of the POS corpora of the respective treebanks used to train the language models described above. This feature template involves three parameters leading to 16 feature subtypes: n-gram order (3- and 5-grams), frequency quartile (1 to 4), POS tag set (original and universal).

`dependency-top-node-pos-tag:` the POS tag of the top node of the dependency tree. The top node is the dependent of the dummy root node. There are two subtypes for this feature template, one for the original POS tag and one for the universal.

[11] All the sentential categories for each language and all WH phrase categories are counted together. For English, sentential categories include `S, SBAR, SINV, SQ, SBARQ` and for French `Sint, Srel, Ssub`. English WH categories include `WHNP, WHPP, WHADVP`.

[12] Language models are trained using SRILM toolkit (`http://www.speech.sri.com/projects/srilm/`) with Witten-Bell smoothing method.

[13] The `NUM` POS tag does not appear in the FTB so there is only one feature in this subtype.

`dependency-top-node-dependent-count:` the number of dependents of the top node

`dependency-top-node-dependency-relations-sequence:` the sequence of all dependency relations which modify the top node

`dependency-top-node-dependents-pos-tag-sequence:` the sequence of the POS tags of the dependents of the top node, including subtypes for original and universal POS tags

`dependency-average-dependent-count:` the average number of dependents of the nodes

`dependency-tree-height:` the height of the dependency tree computed in the same way as the constituency tree height

`dependency-relation-ngrams:` n-gram (3- and 5-grams) sequences of dependency relations of the tokens to their head.

`dependency-relation-count:` the number of most frequent dependency relations in the News training set in the sentence. Since the dependency relation sets of English and French are different, this feature template includes only one feature per dependency relation instead of one for the source and one for the target. There are 16 dependency relations out of 49 in the English side of this training set which appear more than 1000 times. In its French side, there are 10 such relations out of 23. Therefore, this feature template includes 26 binary features.

`dependency-relation-lm-score:` dependency relation n-gram scores against language models trained on the dependency conversions of the respective treebanks for each language. There are two parameters involved in this feature template expanding it into 6 feature subtypes: n-gram order (3- and 5-grams) and LM score type (probability, log probability and perplexity).

`dependency-relation-ngram-quartile-frequency:` the average number of dependency relation n-grams in each n-gram frequency quartile of the dependency relation corpora of the respective treebanks used to train the language models described above. This feature template involves two parameters leading to 8 feature

Table 4.3: QE performance with all hand-crafted syntactic features `SyHC-all` and the reduced feature set `SyHC` on the development set. Statistically significantly better scores compared to their counterpart (same column and the upper row) are in bold.

| Development Set | | | | | | |
|---|---|---|---|---|---|---|
| | BLEU | | 1-TER | | Meteor | |
| | RSME | r | RSME | r | RSME | r |
| `SyHC-all` | 0.1567 | 0.3026 | 0.1851 | 0.2746 | 0.1575 | 0.2996 |
| `SyHC` | **0.1540** | 0.3398 | **0.1819** | **0.3263** | **0.1547** | 0.3452 |

subtypes: n-gram order (3- and 5-grams) and frequency quartile (1 to 4).

`dependency-relation-token-pairs:` pairs of tokens and their dependency relations to their head. This feature template can be considered as the dependency counterpart of the lexical CFG rule.

We build the first hand-crafted syntax-based QE system by combining all of the above features. Table 4.3 shows the performance of this system (`SyHC-all`) on the development and Table 4.5 on the test set. This system outperforms the `B-WMT` baseline on the test set.

There are 311 features in this feature set which expands to 489 features after binarizing the nominal features. Since the feature set is big and also contains many sparse features, we attempt to reduce it through a manual feature selection heuristic.[14] We, in fact, remove the redundant features instead of using exhaustive or other automatic feature selection methods. For example, we investigate whether either the ratio or difference of the source and target numerical features or both of them are redundant. For this purpose, we build three systems, one by removing ratio features, one by removing the difference features and one by removing both of them from the `SyHC-all` feature set. We then compare their performance on the development set and decide which of them is more likely to be useful to keep in the final feature set. This process is also carried out for log probability and perplexity features, original

[14]With regard to the efficiency related to the feature set size, it should be noted that although the kernel computation is faster for hand-crafted features than for tree kernels, we optimise two parameters for the former while only one for the latter. Therefore, due to a considerably larger parameter search grid, tuning for hand-crafted features takes longer, thus the number of features will have noticeable impact on the computation time.

and universal POS-based features, n-gram and language model score features, lexical and non-lexical CFG rules, perplexity and log probability scores, and finally n-gram orders (i.e. 3-gram vs. 5-grams features).

Based on these observations, we remove less useful features from the complete feature set. Our pruning process is efficiency-oriented, i.e. we tend to drop features with small gains. However, in case of a performance drop with the final reduced feature set compared to the complete feature set, we restore such features.

The final reduced feature set contains 104 features which expand to 144 features after binarizing nominal features. The features included in this feature set are listed in Table 4.4.

`SyHC` in Table 4.3 is the system built with the reduced feature set evaluated on the development set. This system performs consistently better than the `SyHC-all` system on the development set, mostly with statistically significant differences as marked in the table. Since our feature reduction process is based on the results on the development set, we use the reduced feature set as our hand-crafted feature set for the rest of the work. However, when applied to the test set, the performance degrades (albeit not statistically significantly) as can be seen in Table 4.5. Considering a more than 70% reduction in feature set size, this relatively small degradation is tolerable.

Compared to `SyTK`, the performances are lower for all MT metrics, though not statistically significantly. It is worth noting that we observed the opposite behaviour on the development set, where the hand-crafted features largely outperform tree kernels. This suggests that the tree kernels are more generalisable, so that the performance drop is smaller on unseen data.

We also combine these features with the WMT 17 baseline features (`B-WMT`). The combined system is `B+SyHC` in Table 4.5. This combination also successfully improves over both syntax-based and baseline systems, confirming the usefulness of syntactic information in collaboration with surface features.

Table 4.4: Features in the reduced feature set

| | |
|---|---|
| | **Constituency** |
| 1 | constituency-tree-root |
| 2 | constituency-tree-height |
| 3 | constituency-tree-size |
| 4 | constituency-tree-size-ratio |
| 5 | constituency-parse-probability |
| 6 | constituency-pseudo-reference-score |
| 7 | constituency-root-cfg-rule-rhs |
| 8 | constituency-lexical-cfg-rules |
| 9 | constituency-nonlexical-cfg-rules |
| 10 | average-constituency-tree-arity |
| 11 | phrase-tag-count |
| 12 | universal-pos-5gram-lm-perplexity |
| 13 | universal-pos-count |
| 14 | first-verb-location |
| | **Dependency** |
| 1 | dependency-top-node-universal-pos-tag |
| 2 | dependency-top-node-dependent-count |
| 3 | dependency-top-node-dependent-ratio |
| 4 | dependency-top-node-dependency-relations-sequence |
| 5 | dependency-top-node-dependents-universal-pos-tag-sequence |
| 6 | dependency-average-dependent-count |
| 7 | dependency-average-dependent-count-ratio |
| 8 | dependency-tree-height |
| 9 | dependency-tree-height-ratio |
| 10 | dependency-relation-3grams |
| 11 | dependency-relation-count |
| 12 | dependency-relation-token-pairs |

### 4.2.4 Combined Syntax-based QE

We combine tree kernels and hand-crafted features to build a full syntax-based QE system. This system is presented in Table 4.6 (SyQE). It improves over both SyTK and SyHC. The improvements for TER and Meteor prediction are slight, but statistically significant for BLEU prediction.

The syntax-based system is combined with B-WMT in B+SyQE. All the gains obtained by the combination are statistically significant, showing again that syntax and surface feature can complement each other.

Table 4.5: QE performance with all hand-crafted syntactic features `SyHC-all` and the reduced feature set `SyHC` on the test set.

| | Test Set | | | | | |
|---|---|---|---|---|---|---|
| | BLEU | | 1-TER | | Meteor | |
| | RSME | r | RSME | r | RSME | r |
| B-WMT | 0.1601 | 0.1766 | 0.1949 | 0.1565 | 0.1625 | 0.2047 |
| SyTK | 0.1581 | 0.2437 | 0.1888 | 0.2774 | 0.1595 | 0.2715 |
| SyHC-all | 0.1603 | 0.2108 | 0.1902 | 0.2510 | 0.1607 | 0.2493 |
| SyHC | 0.1603 | 0.1998 | 0.1913 | 0.2365 | 0.1610 | 0.2516 |
| B+SyHC | **0.1587** | **0.2418** | **0.1899** | **0.2611** | **0.1585** | **0.2964** |

Table 4.6: QE performance with full syntax-based system (`SyQE`) and its combination with WMT baseline features on the News data set

| | BLEU | | 1-TER | | Meteor | |
|---|---|---|---|---|---|---|
| | RSME | r | RSME | r | RSME | r |
| B-WMT | 0.1601 | 0.1766 | 0.1949 | 0.1565 | 0.1625 | 0.2047 |
| SyTK | 0.1581 | 0.2437 | 0.1888 | 0.2774 | 0.1595 | 0.2715 |
| SyHC | 0.1603 | 0.1998 | 0.1913 | 0.2365 | 0.1610 | 0.2516 |
| SyQE | 0.1577 | 0.2535 | 0.1887 | 0.2797 | 0.1594 | 0.2743 |
| B+SyQE | **0.1568** | **0.2802** | **0.1879** | **0.2937** | **0.1576** | **0.3127** |

## 4.3 Syntax-based QE with SymForum Data Set

We now turn our attention to our target domain data set and investigate the application of syntax-based QE on the SymForum data set introduced in Section 3.2.1 of Chapter 3. We build similar QE systems to those in the News data set experiments. Although the general settings in these experiments follow those of the News data set experiments, one difference is in the parsers used.[15] For the English side, we use the same parser but train it on the whole WSJ instead of only its training section. The dependency parse trees are obtained by converting these constituency parse trees. For the French side, we again use the same constituency parser but trained on the entire FTB rather than its training section. For its dependency parsing, we use

[15]We ran these set of experiments at the same time as the semantic-based QE experiments on this data set in Chapter 7. To increase the accuracy of semantic role labelling required for those experiments, we trained these new parses. We thus decided to use these new parses for both experiments for both efficiency and consistency.

Table 4.7: Baseline system performances on SymForum test set

| | 1-HTER | | HBLEU | | Adequacy | | Fluency | |
|---|---|---|---|---|---|---|---|---|
| | RSME | r | RSME | r | RSME | r | RSME | r |
| B-Mean | 0.2442 | - | 0.2907 | - | 0.2501 | - | 0.2796 | - |
| B-WMT17 | 0.2310 | 0.3661 | 0.2696 | 0.3806 | 0.2219 | 0.4710 | 0.2469 | 0.4769 |

ISBN dependency parser (Titov and Henderson, 2007) trained on the dependency conversion of FTB using the Const2Dep converter, instead of applying the converter to the constituency parses as before. This parser requires the POS tagged sentences which are obtained using MElt tagger (Denis and Sagot, 2012) with its FTB-trained built-in model.

We use HBLEU, HTER, Adequacy and Fluency as quality metrics. We first build the baseline systems as in Section 4.2.1 before proceeding to the syntax-based experiments.

### 4.3.1 Baseline QE Systems

Table 4.7 shows the performance of two baseline systems, similar to those used in Section 4.2.1, in predicting HTER, HBLEU, HMeteor, Adequacy and Fluency on the test set of the SymForum data set.[16] B-Mean assigns the mean average of the metric scores in the training set to all test instances. B-WMT17 uses the WMT 17 baseline features. To extract these features, we use slightly different resources than those used in News experiments. Particularly, we combine the English-French Europarl (Koehn, 2005) corpus and the Symantec translation memory described in Section 2.2 to be used for extracting the two translation features (features number 7 and 8 in Table 3.1 in Chapter 3). In addition, the source and target sides of these parallel corpora are used to train the language models needed for extracting n-gram probability features (features number 4 and 5). Furthermore, the source side is used to extract n-gram frequency and percentage features (features number 9 to 15).

[16]As described in Section 3.2.1 of Chapter 3, Adequacy and Fluency scores are scaled to the [0,1] range to be easily comparable to human-targeted scores.

Table 4.8: QE performance with syntactic tree kernels SyTK and their combination with WMT baseline features B+SyTK on SymForum test set.

| | 1-HTER | | HBLEU | | Adequacy | | Fluency | |
|---|---|---|---|---|---|---|---|---|
| | RSME | r | RSME | r | RSME | r | RSME | r |
| B-WMT17 | 0.2310 | 0.3661 | 0.2696 | 0.3806 | 0.2219 | 0.4710 | 0.2469 | 0.4769 |
| SyTK | 0.2267 | 0.3693 | 0.2721 | 0.3559 | 0.2258 | 0.4306 | 0.2431 | 0.5013 |
| B+SyTK | 0.2243 | 0.3935 | 0.2655 | 0.4082 | 0.2215 | 0.4632 | 0.2403 | 0.5144 |

The much higher Pearson r for human-targeted metric prediction obtained here using B-WMT17 compared to those on the News data set is notable in Table 4.7. Though not directly comparable, this may be due to the fact that these metrics are a better indicator of the translation quality when used via human-targeted scoring method rather than using pre-translated references, thus easier to learn. Another interesting observation is that manual metric predictions achieve much higher Pearson r than human-targeted metric predictions. This can partially be attributed to the score distribution as there are only 5 different scores to learn in manual metric prediction. Nevertheless, another reason may be that manual metric scores better capture the translation quality, and thus, are easier to learn and predict.

Note that in subsequent mentions of the baseline system, we will be referring to the B-WMT17 system.

### 4.3.2 Syntax-based QE with Tree Kernels

Using the same setting as in Section 4.2.2, we build tree kernel QE systems to predict all five metrics for the SymForum data set. The performances of these systems are shown in Table 4.8.

Unlike with the News data set, the syntactic tree kernels (SyTK) do not seem to be better than the B-WMT17 baseline. They outperform the baseline in predicting HTER and Fluency but the gaps are not statistically significant. On the other hand, in predicting BLEU and Adequacy, they perform worse than the baseline, though only the latter difference is statistically significant.

Table 4.9: QE performance with hand-crafted features SyHC and their combination with WMT baseline features B+SyHC on SymForum test set

| | 1-HTER | | HBLEU | | Adequacy | | Fluency | |
|---|---|---|---|---|---|---|---|---|
| | RSME | r | RSME | r | RSME | r | RSME | r |
| B-WMT17 | 0.2310 | 0.3661 | 0.2696 | 0.3806 | 0.2219 | 0.4710 | 0.2469 | 0.4769 |
| SyTK | 0.2267 | 0.3693 | 0.2721 | 0.3559 | 0.2258 | 0.4306 | 0.2431 | 0.5013 |
| SyHC | 0.2435 | 0.2572 | 0.2797 | 0.3080 | 0.2334 | 0.3961 | 0.2479 | 0.4696 |
| B+SyHC | 0.2265 | 0.4159 | 0.2689 | 0.4080 | 0.2221 | 0.4795 | 0.2387 | 0.5269 |

When these tree kernels are combined with the baseline features (B+SyTK in Table 4.8), improvements are seen over the best of the two systems except for Adequacy prediction. The only statistically significant improvement however is the one for HTER prediction. Again, this is not on a par with the results we saw with the News data set, where the combination led to significant improvements across the board.

### 4.3.3 Syntax-based QE with Hand-crafted Features

In Section 4.2.3, we designed a set of hand-crafted features extracted from constituency and dependency trees of the source and target side of the News data set. We now extract the same set of features used for the experiments on that data set from the SymForum data set and build a QE system using the same SVM setting. It should be noted that despite using the same feature templates, the actual features in the feature set (thus the feature set size) are slightly different from the News data set due to different frequency threshold cutoff values for binarized nominal features tuned for this data set. Specifically, there are 127 features in the feature set compared to 144 features for the case of the News data set. Table 4.9 displays the performance of the system built using this feature set (SyHC).

As can be seen, the scores are substantially lower compared to the tree kernel system (SyTK in the same table). Despite all the big gaps, they are not statistically significant for Pearson r of the HBLEU and Adequacy prediction and RMSE of the Fluency prediction. The differences are particularly pronounced for the case of HTER prediction with more than 11 Pearson r points. Similarly, the system is

Table 4.10: QE performance with full syntax-based system (SyQE) and its combination with WMT baseline features on the SymForum data set

| | 1-HTER | | HBLEU | | Adequacy | | Fluency | |
|---|---|---|---|---|---|---|---|---|
| | RSME | r | RSME | r | RSME | r | RSME | r |
| B-WMT17 | 0.2310 | 0.3661 | 0.2696 | 0.3806 | 0.2219 | 0.4710 | 0.2469 | 0.4769 |
| SyTK | 0.2267 | 0.3693 | 0.2721 | 0.3559 | 0.2258 | 0.4306 | 0.2431 | 0.5013 |
| SyHC | 0.2435 | 0.2572 | 0.2797 | 0.3080 | 0.2334 | 0.3961 | 0.2479 | 0.4696 |
| SyQE | 0.2255 | 0.3824 | 0.2711 | 0.3650 | 0.2248 | 0.4393 | 0.2419 | 0.5087 |
| B+SyQE | 0.2236 | 0.4017 | 0.2686 | 0.3852 | 0.2219 | 0.4632 | 0.2391 | 0.5255 |

outperformed by the baseline, where all the differences except for Fluency prediction are remarkable and statistically significant.

These features can nevertheless be used to boost the performance of the baseline features. This can be seen in the performance of B+SyHC in Table 4.9, which is the combination of these two systems. All the scores are better than the baseline scores, with the differences being statistically significant for HTER and Fluency prediction.

### 4.3.4 Combined Syntax-based QE

Combining tree kernels and hand-crafted features in Section 4.2.4 led to slight improvement for the News data set. We similarly combine SyTK and SyHC to build our full syntax-based QE system, SyQE, with the SymForum data set. The performance of this system is shown in Table 4.10.

Compared to SyTK, the better of the two component systems, all the scores have increased, the biggest improvement obtained for HTER prediction. Interestingly, the small differences are statistically significant. Moreover, the combination seem to be more useful for this data set than the News data set.

Compared to the baseline, we see mixed results: HTER and Fluency prediction scores are higher than the baseline but BLEU and Adequacy prediction scores are lower. None of these differences are statistically significant. This is again unlike what we observed with the News data set, suggesting that the performance of syntax-based QE is dependent on the data set. One of the differences between the News and the

SymForum data set is that the syntactic information for the latter is expected to be noisier due to out-of-domain parsing as well as parsing erroneous text. This may consequently affect the performance of the downstream QE task. In the next chapter, we further investigate this hypothesis by elaborating the effect of parse quality on syntax-based QE.

Finally, we combine the full syntax-based system with the baseline features. The resulting system is `B+SyQE` presented in Table 4.10. According to the results, the combination further improves the HTER and Fluency prediction statistically significantly. It is also able to cover the gap between the syntax-based system and the baseline in BLEU and Adequacy prediction, but no improvement over the baseline is achieved.

An interesting point to note so far is that HTER and Fluency prediction tend to behave similarly and so do BLEU and Adequacy prediction. Additionally, the former pair tend to benefit more from syntactic information. This is expected for the case of fluency measure as it is obviously concerned with the grammar of the MT output. For the case of HTER, this suggests that this metric also captures the grammaticality of the translation.

## 4.4 Summary and Conclusion

We built a set of quality estimation systems which relied solely on syntactic information derived from the machine translation source and target. The syntactic information we used were derived from both constituency and dependency parses, each capturing a different aspect of language syntax. We evaluated our approaches using two different data sets: 1) the News data set, a set of edited sentences in the newswire domain which come from the WMT13 News development data set, the purpose of which was to preclude the extra level of noise from out-of-domain parsing affecting the conclusions, and 2) the SymForum data set, a set of user-generated sentences in the security software domain which come from Symantec Norton fo-

rums, the target domain text of this thesis.

The quality estimation systems were based upon two approaches to incorporating syntactic information in a machine learning framework for quality estimation: 1) tree kernels and 2) hand-crafted features. These approaches were experimented with both separately and in combination with each other. The results show that tree kernel are more effective in learning translation quality scores from parse trees than the hand-crafted features, since they are exposed to the entire tree structure from which they can extract the most useful information for this purpose.

We compared the performance of the syntax-based QE systems with a baseline system built using the well known surface features used as a reasonable baseline in quality estimation shared tasks of the WMT workshop. According to the results, the syntax-based systems perform substantially better than the WMT baseline system when applied to the News data set. In addition, these systems were successfully combined, the resulting system being significantly better than both components. However, inconsistent results were observed with the SymForum data set, where the syntax-based systems were outperformed by the baseline in BLEU and Adequacy prediction, and the combining the systems did not improve the Adequacy prediction. Although these differences were not statistically significant, they suggest that the QE performance using syntactic parses is dependent on the data and score distribution. Since one of the differences between these two data sets is the lower quality of the parsing of SymForum data due to out-of-domain parsing, this behaviour can be attributed to the parse quality. We investigate this hypothesis in the next chapter.

In sum, syntactic information is valuable for quality estimation both when used alone and when combined with other sources of information. In the next chapter we elaborate further on the role of syntax in estimating the quality of machine translation.

# Chapter 5

# In-depth Analysis of Syntax-based Quality Estimation

As explained in Section 3.1 of Chapter 3, syntactic information has mainly been used in conjunction with other types of information with the aim of building strong quality estimation systems, but none of the previous work has examined in detail the use of syntax in QE for MT. In this chapter, we conduct an in-depth study of the role of syntax in QE from various perspectives based on the experiments from the previous chapter.

The syntactic parses obtained using the automatic parsers are not free of errors, as the state-of-the-art parsing performance is still far below perfect. Moreover, this state-of-the-art is for parsing the edited newswire text to which the parsers are tailored. These parsers are applied to the output of machine translation and the user-generated content of the Norton forum, both of which can be grammatically ill-formed. The parsers furthermore face new vocabulary, syntactic constructions and writing style in parsing the forum text. These situations are known to impede the parsing performance leading to noisy parses (Foster et al., 2011a). This raises an interesting question: *To what extent is QE for MT influenced by the quality of the syntactic information provided to it, i.e. does the accuracy of the underlying parsing system used to provide the syntactic features influence the accuracy of the*

*QE system?* We seek to answer this question by comparing QE systems which use syntactic parses of varying quality. The parsing quality is varied by reducing the size of the parser training data. The results can shed more light on the problem observed in the previous chapter that the syntax-based QE was more useful for the News data set than for the SymForum data set.

As stated in the previous chapter, syntactic information can be useful in quality estimation by capturing the structural complexity of the source sentence and the grammaticality of its machine translation. In order to identify the contribution of each of these factors, we decompose the QE systems built in the previous chapter to its source and target components. Besides, the performance of the target component may be affected by the fact that it is built upon parses of machine translation output which is generally expected to contain grammatical problems. To better understand the effect of parsing machine-translated sentences in syntax-based QE, we build the same QE systems in the opposite translation direction. We then decompose the new QE systems into their source and target components again and study the resulting behaviours.

Based on our findings from these experiments, we design a set of heuristics to modify the French parse trees by adding more structures to them so that the new parse trees become more useful in the syntax-based QE systems.

In the rest of this chapter, we first review the related work in Section 5.1. In Section 5.2, we examine the impact of parser accuracy on the performance of the syntax-based QE systems using both News and SymForum data sets. In Section 5.3, the role of source and target syntax in QE is verified. Section 5.4 describes the heuristics designed to modify the French parse trees and the experiments replicated using the modified trees.[1]

[1]For easy comparison, all the experimental results in this chapter are also presented in Tables A.1 and A.2 in Appendix A, besides other QE results, for the SymForum and News data sets respectively.

## 5.1 Related Work

There have been relatively few attempts to investigate the impact of parser accuracy in downstream applications. Quirk and Corston-Oliver (2006) report that a syntax-enhanced MT system is sensitive to a decrease in parser quality obtained by training the parser on smaller training sets. However, the BLEU score improvement they achieve seems small compared to the underlying parser accuracy difference. On the other hand, Zhang et al. (2010) experiment with a different syntax-enhanced MT system and do not observe a difference between the performance of MT systems using parses of different quality.

Johansson and Nugues (2007) introduce a constituency-to-dependency converter and find that syntactic dependency trees produced using this converter yield more accurate semantic role labels than syntactic dependency trees produced using a less sophisticated converter despite the fact that trees produced using the older converter tend to have higher attachment scores than the trees produced using the new converter. Mollá and Hutchinson (2003) find significant differences between two dependency parsers in a task-based evaluation involving an answer extraction system but bigger differences between the two parsers when evaluated intrinsically.

Miyao et al. (2008) and Goto et al. (2011) evaluate a suite of state-of-the-art English statistical parsers on the tasks of protein-pair interaction extraction and patent translation respectively, and find only small (albeit sometimes statistically significant) differences between the parsing systems in those tasks. Perhaps this is not surprising as the parsing systems also perform similarly when evaluated intrinsically.

Our investigation of the impact of parser accuracy is closest to that of Quirk and Corston-Oliver (2006) since we are taking one type of parsing model and comparing different instantiations of this model which differ substantially with respect to their intrinsic evaluation scores. However, like Miyao et al. (2008) and Goto et al. (2011), we also compare the performances of models which have similar intrinsic evaluation scores but produce different output.

Table 5.1: Parser $F_1$s for various training set sizes: the sizes in bold are selected for the experiments.

| | English | | | | | French | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Training size | 100 | **1K** | 10K | 20K | **40K** | 100 | **500** | 2.5K | 5K | **10K** |
| $F_1$ | 51.06 | 72.53 | 87.69 | 88.47 | 89.55 | 52.85 | 66.51 | 78.55 | 81.85 | 83.40 |

## 5.2 Parser Accuracy in Syntax-based QE

In order to investigate the effect of parsing accuracy in syntax-based quality estimation built using those parses, we train two parsing models – one "higher-accuracy" model and one "lower-accuracy" model – for each language. We use training set size to control the accuracy. The higher-accuracy models are the ones used so far for parsing our data. For the lower-accuracy models, we first select four random subsets of varying sizes from the larger training sets for each language[2] and measure the performance of the resulting models on the standard parsing test sets[3] using Parseval $F_1$ as shown in Table 5.1.[4]

The worst-performing models for each language are those trained on 100 training sentences. However, these models fail to parse about 10 and 2 percent of our English and French data respectively. Since the failed sentences are not necessarily parallel in the source and target sides, this could affect the downstream QE performance. Therefore, we opt to employ as our "lower-accuracy" models the second smallest training set sizes, which are 1K sentences for English and 500 for French. For both languages, the difference in $F_1$ between the lower-accuracy and higher-accuracy models is about 17 points. In order to measure how different the parses produced by these models are on our QE data, we compute their $F_1$ relative to each other. The $F_1$ for the English model pair is 71.50 and for French 63.19.

In the following sections, we apply these parsing models on the News and Sym-Forum data sets and rebuild the QE systems.

[2]Each smaller subset is contained in all the larger subsets.
[3]WSJ Section 23 and the FTB test set.
[4]All parsing models are trained with 5 split/merge cycles.

Table 5.2: QE performance with systems built using parses of higher- ($_H$ subscripts) and lower-accuracy ($_L$ subscripts) parsing models

| | BLEU | | 1-TER | | Meteor | |
|---|---|---|---|---|---|---|
| | RSME | r | RSME | r | RSME | r |
| SyQE$_H$ | 0.1577 | 0.2535 | 0.1887 | 0.2797 | 0.1594 | 0.2743 |
| SyQE$_L$ | 0.1583 | 0.2341 | 0.1887 | 0.2796 | 0.1600 | 0.2606 |
| SyTK$_H$ | 0.1581 | 0.2437 | 0.1888 | 0.2774 | 0.1595 | 0.2715 |
| SyTK$_L$ | 0.1583 | 0.2350 | 0.1888 | 0.2792 | 0.1600 | 0.2620 |
| SyHC$_H$ | 0.1603 | 0.1998 | 0.1913 | 0.2365 | 0.1610 | 0.2516 |
| SyHC$_L$ | 0.1609 | 0.1750 | 0.1914 | 0.2262 | 0.1613 | 0.2336 |

### 5.2.1 The News Data Set

We first investigate the impact of the intrinsic quality of the parse trees on the QE system using the News data set. We build a similar QE system to SyQE in Section 4.2.4 of the previous chapter, but with the parse trees of the lower-accuracy parsing models. Table 5.2 shows the performances of these two QE systems (SyQE$_H$ and SyQE$_L$). To distinguish between the systems, the subscripts $_H$ and $_L$ are added to the system names to represent higher and lower accuracy parsing models respectively.

Surprisingly, no statistically significant difference is seen between the performance of the systems built with two different parse qualities. On TER prediction, both systems perform the same.

To better understand the behaviour of these systems, we break them down into their components: tree kernels vs. hand-crafted features, constituency vs. dependency features, source side vs. target side features. SyTK$_H$ and SyTK$_L$ in Table 5.2 are the tree kernel components and SyHC$_H$ and SyHC$_L$ the hand-crafted components. As the results suggest, the tree kernel and hand-crafted components behave similarly with respect to the parser accuracy difference, since none of the differences are large or statistically significant. This is particularly visible for tree kernels.

We now study the tree kernel and hand-crafted systems separately in terms of their components. Tables 5.3 and 5.4 present the performances of these components built using the higher- and lower-accuracy parses. C, D, S and T are constituency,

Table 5.3: Tree kernel QE systems with higher- and lower-accuracy trees (C: constituency, D: dependency, ST: Source and Translation, $_H$: Higher-accuracy parsing model, $_L$: Lower-accuracy parsing model)

| | BLEU | | 1-TER | | Meteor | |
|---|---|---|---|---|---|---|
| | RSME | r | RSME | r | RSME | r |
| SyTK/C-ST$_H$ | 0.1584 | 0.2307 | 0.1896 | 0.2641 | 0.1594 | 0.2748 |
| SyTK/C-ST$_L$ | 0.1582 | 0.2348 | 0.1890 | 0.2733 | 0.1596 | 0.2698 |
| SyTK/D-ST$_H$ | 0.1591 | 0.2103 | 0.1907 | 0.2412 | 0.1616 | 0.2213 |
| SyTK/D-ST$_L$ | 0.1597 | 0.1902 | 0.1913 | 0.2279 | 0.1623 | 0.2025 |
| SyTK/C-S$_H$ | 0.1583 | 0.2312 | 0.1904 | 0.2521 | 0.1590 | 0.2824 |
| SyTK/C-S$_L$ | 0.1582 | 0.2335 | 0.1901 | 0.2554 | 0.1599 | 0.2638 |
| SyTK/C-T$_H$ | 0.1608 | 0.1479 | 0.1925 | 0.2018 | 0.1620 | 0.2124 |
| SyTK/C-T$_L$ | 0.1616 | 0.1204 | 0.1934 | 0.1800 | 0.1632 | 0.1773 |
| SyTK/D-S$_H$ | 0.1598 | 0.1869 | 0.1925 | 0.2004 | 0.1630 | 0.1832 |
| SyTK/D-S$_L$ | 0.1601 | 0.1780 | 0.1933 | 0.1816 | 0.1630 | 0.1835 |
| SyTK/D-T$_H$ | 0.1598 | 0.2102 | 0.1916 | 0.2204 | 0.1622 | 0.2051 |
| SyTK/D-T$_L$ | 0.1604 | 0.1679 | 0.1924 | 0.2037 | 0.1628 | 0.1867 |

dependency, source side and target side components respectively.

SyTK/C-ST$_H$ and SyTK/C-ST$_L$ in Table 5.3 are the tree kernel systems with the constituency trees of both source and translation with higher- and lower-accuracy parsing models respectively. As can be seen, SyTK/C-ST$_L$ achieves even better scores in BLEU and TER prediction. In Meteor prediction, the difference is the opposite but not significant. On the other hand, the dependency component is negatively affected by the lower quality of the parse trees but the gaps are neither large nor statistically significant (SyTK/D-ST$_H$ vs. SyTK/D-ST$_L$).

We now further split these systems into source and translation sides. SyTK/C-S$_H$ and SyTK/C-S$_L$ use the higher- and lower-accuracy constituency trees of only the source text. The behaviour is similar to when constituency trees of both sides are used (SyTK/C-ST$_H$ and SyTK/C-ST$_L$). However, the system using higher-accuracy constituency trees of the target (SyTK/C-T$_H$) achieves better scores than the one using the lower-accuracy ones (SyTK/C-T$_L$), but, again, this difference is not statistically significant.

SyTK/D-S$_H$ and SyTK/D-S$_L$ are the systems using the dependency trees of only the source text. The higher-accuracy system better predicts BLEU and TER but not

Table 5.4: Hand-crafted QE systems with higher- and lower-accuracy trees (C: constituency, D: dependency, ST: Source and Translation, $_H$: Higher-accuracy parsing model, $_L$: Lower-accuracy parsing model)

| | BLEU | | 1-TER | | Meteor | |
|---|---|---|---|---|---|---|
| | RSME | r | RSME | r | RSME | r |
| SyHC/C-ST$_H$ | 0.1604 | 0.2046 | 0.1906 | 0.2443 | 0.1604 | 0.2538 |
| SyHC/C-ST$_L$ | 0.1613 | 0.1739 | 0.1894 | 0.2663 | 0.1599 | 0.2643 |
| SyHC/D-ST$_H$ | **0.1617** | **0.1593** | 0.1956 | 0.1609 | 0.1634 | 0.1813 |
| SyHC/D-ST$_L$ | 0.1125 | 0.1633 | 0.1944 | 0.1491 | 0.1638 | 0.1535 |
| SyHC/C-S$_H$ | 0.1613 | 0.1748 | 0.1930 | 0.1874 | 0.1621 | 0.2115 |
| SyHC/C-S$_L$ | 0.1616 | 0.1652 | 0.1933 | 0.1803 | 0.1620 | 0.2113 |
| SyHC/C-T$_H$ | 0.1622 | 0.1585 | 0.1936 | 0.1801 | 0.1622 | 0.2116 |
| SyHC/C-T$_L$ | 0.1624 | 0.1409 | 0.1935 | 0.1747 | 0.1630 | 0.1861 |
| SyHC/D-S$_H$ | 0.1385 | 0.1624 | 0.1939 | 0.1616 | 0.1626 | **0.1932** |
| SyHC/D-S$_L$ | 0.1381 | 0.1625 | 0.1946 | 0.1466 | 0.1636 | 0.1597 |
| SyHC/D-T$_H$ | **0.1643** | **0.0583** | 0.1983 | 0.0247 | 0.1659 | 0.0979 |
| SyHC/D-T$_L$ | 0.1720 | -0.0282 | 0.1978 | -0.0032 | 0.1655 | 0.0567 |

Meteor. The gaps are however, not statistically significant. On the other hand, on the target side higher-accuracy dependency trees in SyTK/D-T$_H$ perform consistently better than their lower-accuracy counterpart (SyTK/D-T$_L$), especially pronounced in BLEU prediction. Although the gap on BLEU prediction here is the only large difference observed among all settings, it is surprisingly not statistically significant.[5]

Putting all the above observations together, the large performance gap between two parsing models does not affect the performance of the tree-kernel-based QE system. This is especially visible for the constituency tree kernels and for the parse trees of the source (English) side. On the other hand, a small effect is seen with the parse trees of the target (French) side, especially with dependency trees. However, since source-side constituency trees contribute the major part of the performance, this effect fades out in the full system.

For the hand-crafted system, the performances of the components are shown in Table 5.4. Compared to the tree kernel components, the extent of gaps is bigger

[5]The high scores of SyTK/D-T$_H$ seem to be happening by chance, because on the development set, on which the parameters are tuned, the scores are much lower. RMSE is 0.1633 and Pearson r is 0.1355. The same applies to SyTK/D-T$_L$ where RMSE is 0.1621 and Pearson r is 0.1175. Not only is the Pearson r gap not statistically significant, the RMSE of SyTK/D-T$_L$ is even better.

as can also be seen in the three statistically significant differences marked in bold. Similar to tree kernels, most of the parse quality impact seen on the performance of QE with hand-crafted features (SyHC) is due to the dependency trees and target (French) side trees.[6] In addition, the quality of source (English) side trees is more effective in the performance of the hand-crafted systems. In the next section, we further validate these results using low accuracy parses obtained in a different way.

#### 5.2.1.1 Analysing the Results

One may argue that the way the parser accuracy is varied here could impact the results – a parser with similar $F_1$ but different output may lead to a different conclusion. It is possible to test this by using the parsing model from a lower split/merge (SM) cycle of our iterative PCFG-LA parser. For example, the models from the first SM cycle with a 10K training set size for English and a 2.5K training set size for French score 73.04 and 70.22 $F_1$ points on their respective test sets. While these scores are close to those of the lower-accuracy models used above (72.53 and 66.51), their outputs are different: the parses with the two lower-accuracy English models achieve only 66.46 $F_1$ against each other and with the two French ones 66.51 $F_1$. We use the parse trees of these alternative lower-accuracy parsing models to build a new tree kernel QE system. The RMSE and Pearson r for BLEU prediction are 0.1585 and 0.2316. These scores are not statistically significantly different compared to $\texttt{SyTK}_H$ in Table 5.2 (15.81 and 24.37), strengthening our conclusion that intrinsic parse accuracy is not crucial for QE of MT.

Another question is to what extent we require a linguistically realistic syntactic structure which retains some form of regularity no matter how accurate. To answer this question, we build random tree structures for source and translation segments. The random tree for a segment is generated by recursively splitting the sentence into random phrases and randomly assigning them a syntactic label.[7] We parse

[6]Note however that the systems built with target side dependency trees (SyHC/D-T) perform very poorly especially in predicting BLEU and TER. This can be the reason for the performance degradation when constituency and dependency trees are combined in SyHC.

[7]The English random model achieves an $F_1$ of around 0.5 and the French model an $F_1$ of 0.2.

the source and translation segments using this method and build a tree kernel QE system with the output trees. The RMSE and Pearson r are 0.1631 and -0.0588 respectively. This shows that tree kernels still require the regularity encoded in the lower- and higher-accuracy trees and perhaps this regularity exists in the output of low accuracy parsers, regardless of its quality.

### 5.2.2 The SymForum Data Set

The experiments on the News data set in the previous section showed that the intrinsic accuracy of the parse trees did not affect the accuracy of the syntax-based quality estimation built with those trees. It is interesting to know how that finding applies to another data set with different characteristics. We therefore rebuild the `SyQE` system of Section 4.3.4 of the previous chapter using parsers with lower accuracy.

For the English side and the constituency parsing of the French side, we use the same low-accuracy parsers as in the previous section. For the dependency parsing of the French side, however, we need to train a new parser as the parser used with this data set is different. Using a similar strategy, we build such a parser by training `ISBN` on the same fraction of the dependency conversion of FTB as used in Section 5.2.1, i.e. 500 parse trees. Table 5.5 compares the performance of this parsing model evaluated on the test section of FTB with the performance of a higher-accuracy model, which is trained on the training section of FTB. The performance is measured by unlabelled and labelled attachment scores (UAS and LAS). As shown in the table, there are more than 9 LAS and 7.5 UAS points difference between the parsers. Note that this higher accuracy parser is not exactly the one used for parsing the SymForum data as we needed to leave the test section for evaluation.

Using the output of these lower-accuracy parsers, we build a full syntax-based system named $\texttt{SyQE}_L$ as a counterpart to the `SyQE` in Table 4.10 in Chapter 4. Table 5.6 shows the performance of both systems. According to the results, $\texttt{SyQE}_H$, which is the same system as the `SyQE` in Table 4.10, outperforms $\texttt{SyQE}_L$ in predicting

Table 5.5: UAS and LAS of lower- and higher-accuracy French dependency parsers

| | UAS | LAS |
|---|---|---|
| $\texttt{ISBN}_H$ | 0.8892 | 0.8646 |
| $\texttt{ISBN}_L$ | 0.8132 | 0.7753 |

Table 5.6: QE performance with systems built using parses of higher- ($_H$ subscripts) and lower-accuracy ($_L$ subscripts) parsing models

| | 1-HTER | | HBLEU | | Adequacy | | Fluency | |
|---|---|---|---|---|---|---|---|---|
| | RSME | r | RSME | r | RSME | r | RSME | r |
| $\texttt{SyQE}_H$ | 0.2255 | 0.3824 | 0.2711 | 0.3650 | 0.2248 | 0.4393 | 0.2419 | 0.5087 |
| $\texttt{SyQE}_L$ | 0.2273 | 0.3647 | 0.2731 | 0.3455 | 0.2249 | 0.4386 | 0.2415 | 0.5097 |

HTER and HBLEU by around 2 Pearson r points. These differences are however not statistically significant. On the other hand, the two systems achieve a similar performance in predicting manual metric scores. The results suggest that, similar to what we observed with the News data set in Section 5.2.1, there is no correlation between intrinsic accuracy of the parses and the accuracy of QE systems accuracy built with them. This finding also rules out the possibility of the hypothesis made in Section 4.4 of the previous chapter that the syntax is less useful in quality estimation of forum text translation due to additional noise in its parsing. However, it can still be related to the aspects of the parsing quality which is not captured by PARSEVAL metric. For example, the parsing inconsistency within the document due to structural inconsistency of the forum text can be a reason affecting the performance of the syntax-based QE system on this data set.

## 5.3 Source and Target Syntax in QE

Although we did not find a significant difference between QE systems built using high- and low-accuracy parses in the previous sections, the results of the tree kernel system breakdown in Table 5.3 reveals a performance gap from a different angle. It can be seen that, the source side (English) parser trees perform substantially better

Table 5.7: Tree kernel QE performances on the News data set with only source (English) or translation (French) side trees (S: source, T: translation)

| | BLEU | | 1-TER | | Meteor | |
|---|---|---|---|---|---|---|
| | RSME | r | RSME | r | RSME | r |
| SyTK/CD-S | 0.1584 | 0.2294 | 0.1899 | 0.2573 | 0.1596 | 0.2690 |
| SyTK/CD-T | 0.1597 | 0.2101 | 0.1913 | 0.2270 | 0.1613 | 0.2299 |
| SyTK/C-S | 0.1583 | 0.2312 | 0.1904 | 0.2521 | 0.1590 | 0.2824 |
| SyTK/C-T | 0.1608 | 0.1479 | 0.1925 | 0.2018 | 0.1620 | 0.2124 |
| SyTK/D-S | 0.1598 | 0.1869 | 0.1925 | 0.2004 | 0.1630 | 0.1832 |
| SyTK/D-T | 0.1598 | 0.2102 | 0.1916 | 0.2204 | 0.1622 | 0.2051 |

than the target (French) parse trees (e.g. SyTK/C-S$_H$ vs. SyTK/C-T$_H$, SyTK/C-S$_L$ vs. SyTK/C-T$_L$). In order to investigate this difference, we take a closer look at the effect of parse trees of different languages as well as parse trees of machine translation input (well-formed text) versus machine translation output (possibly ill-formed text), in syntax-based quality estimation using tree kernels on the News data set. We decompose the tree kernel system into the source and target trees. Table 5.7 depicts the performance of these systems. SyTK/CD-S is the system with constituency and dependency trees of the source and SyTK/CD-T the system with those of the target.

At a glance, it can be seen that the tree kernels of the source perform better than the tree kernels of the target However, when the systems are further broken down into constituency and dependency tree kernels (SyTK/C-S, SyTK/C-T, SyTK/D-S, SyTK/D-T), this is only true for the constituency tree kernels, albeit with much bigger gaps. On the other hand, the dependency trees of the target are more useful than the source dependency trees.

The large difference between SyTK/C-S and SyTK/C-T could be attributed to the (presumably) lower quality of the parse trees of the target text (see Figure 1). Although this low quality is expected to affect the dependency parse trees in the same way as they are directly derived from the consistency trees, it however is not the case according to the results. Perhaps the problematic aspects of the MT parses are only reflected in the constituency trees and are abstracted away from the

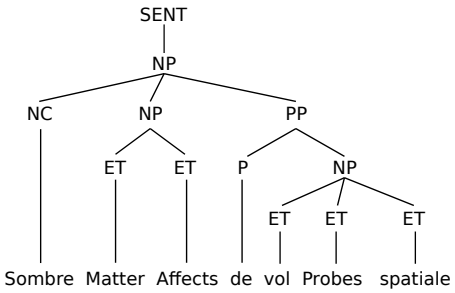



Figure 5.1: Parse tree of the machine translation of *Dark Matter Affects Flight of Space Probes* to French

dependency trees. The second hypothesis is that this large gap can rather be due to the idiosyncrasies of the underlying treebanks which is not carried over via the conversion tools to the dependency structure.

To separate out the role of treebank and machine translation in the behaviours observed above, we switch the translation direction to French-to-English. Therefore, we now parse the well-formed French input sentences and the machine-translated English segments.[8] If the first hypothesis were true, the target side parse trees in this direction would still underperform the source side ones. Otherwise, if the second hypothesis were correct, French parse trees would still be less useful even in the source side.

Table 5.8 shows the results for all tree kernel QE systems for the French-English translation direction. As the magnitude of the scores is very similar to the English-French direction, comparison will be easier.[9] According to the results, all the systems using target trees outperform those using source trees. The difference between `SyTK-FE/CD-T` and `SyTK-FE/CD-S` and between `SyTK-FE/C-T` and `SyTK-FE/C-S` are especially substantial and statistically significant.

Therefore, it is apparent that the suspected lower quality of constituency parse trees of MT output is not the reason for lower performance of the quality estimation using kernels derived from them. Rather, parse trees of French sentences seem to

[8]Note that segments are identical to the English-French direction and are translated and parsed using the same systems described earlier.

[9]We do not predict Meteor in these settings, as scoring with Meteor in the French-English direction requires parameter weight tuning.

Table 5.8: Tree kernel QE performances for French-English direction on the News data set (FE: French to English, C: constituency, D: dependency, S: source, T: translation)

| | BLEU | | 1-TER | |
|---|---|---|---|---|
| | RSME | r | RSME | r |
| SyTK-FE | 0.1561 | 0.2334 | 0.1955 | 0.2897 |
| SyTK-FE/CD-S | 0.1574 | 0.1830 | 0.1986 | 0.2339 |
| SyTK-FE/CD-T | 0.1559 | **0.2423** | 0.1961 | 0.2803 |
| SyTK-FE/C-S | 0.1581 | 0.1578 | 0.2008 | 0.1903 |
| SyTK-FE/C-T | **0.1556** | **0.2336** | 0.1979 | **0.2486** |
| SyTK-FE/D-S | 0.1577 | 0.1655 | 0.1981 | 0.2453 |
| SyTK-FE/D-T | 0.1579 | 0.1886 | 0.1984 | 0.2387 |

be less useful than the English ones in such a framework.

One reason could be the relatively lower performance of the French parsing compared to English (see Table 5.1). However, we showed in Section 5.2.1 that parser accuracy as measured by Parseval F-score does not appear to affect the quality of downstream QE. Therefore, we seek the answer in the difference between the annotation scheme of English Penn Treebank (PTB) and French Treebank (FTB). FTB is known to have a relatively flatter structure (Schluter and van Genabith, 2007). For example, it lacks a verb phrase (VP) node and phrases modifying the verb are the sibling of the verb nucleus. In the next section, we examine the effect of this characteristic of French parse trees on the quality estimation systems built using those trees.

## 5.4 Modifying French Parse Trees

In order to check whether the annotation strategy is a reason for the lower performance of French constituency tree kernels, we apply a set of three heuristics which introduce more structure to the French parse trees (1&2) or simply make them more PTB-like (3):

- *Heuristic 1* automatically adds a VP node above the verb node (VN) and at most 3 of its immediate adjacent nodes if they are noun or prepositional phrases

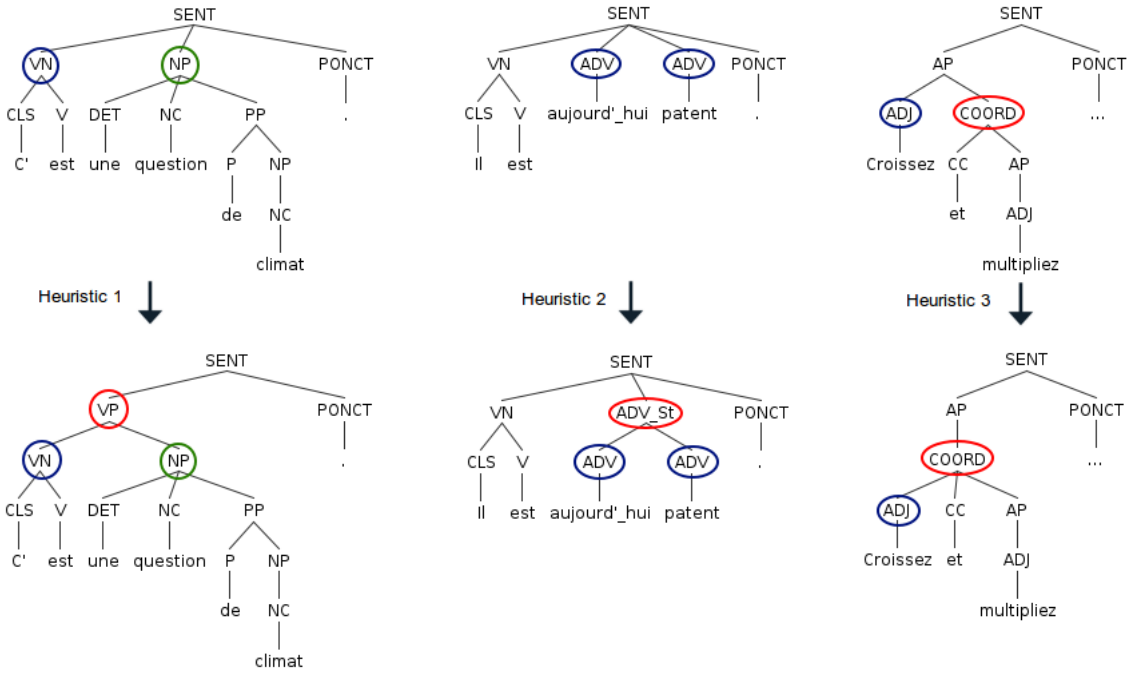



Figure 5.2: Application of tree modification heuristics on example French translation parse trees

(`NP` or `PP`).

- *Heuristic 2* stratifies some of the production rules in the tree by grouping together every two identical adjacent POS tags under a new node with a tag made of the POS tag suffixed with `_St`.
- *Heuristic 3* moves coordinated nodes (the immediate left sibling of the `COORD` node) under `COORD`.

Figure 5.2 shows examples of the application of each of these methods. These heuristics are applied to the parses of the News and SymForum data in the following sections and their effect on the resulting QE systems are examined.

### 5.4.1 The News Data Set

We first apply these heuristics to the parsed MT output of the News data set in the English-French translation direction and rebuild the tree kernel system with target constituency trees (`SyTK/C-T`) and the full tree kernel system (`SyTK/CD-ST`) with the modified trees. The results are presented in Table 5.9.

Despite the possibility of introducing linguistic errors, these heuristics yield a statistically significant improvement in QE performance for all settings except for TER prediction with SyTK/C-T which largely degrades.[10] Unsurprisingly, the changes are bigger for the system with only target constituency trees as there are three other tree types involved in the full system (SyTK/CD-ST). It is nonetheless surprising that the TER prediction degrades with modified French constituency trees when used alone but improves when combined with other trees. These results suggest that the structure of the French constituency trees can be a factor in the lower performance of its tree kernels in QE.[11]

The gain achieved by applying these heuristics is related to the fact that there are more similar fragments extracted from the modified structure which are useful for the tree kernel system. For example, in the original top left tree in Figure 5.2, there is no chance that a fragment consisting only of VN and NP – a very common structure and thus useful in calculating tree similarity – will be extracted by the *subset* tree kernel. The reason is that this kernel type does not allow the production rule to be split (in this case the rule expanding the S node). However, after applying Heuristic 1, the fragment equivalent to VP -> VN NP production rule can be easily extracted.

Among the three heuristics, the first one contributes the largest part of the improvement; the other two have a very slight effect according to the results of their individual application, though they contribute to the overall performance when all three are combined. There are 12,060 VP nodes and 2059 _St nodes in the whole data set after the application of Heuristic 1 and 2 respectively. There are also 3276 COORD nodes in the data set. This difference could explain the performance gap between the first and the other two heuristics.

In analysing the degradation observed in TER prediction, we found only Heuris-

[10] Despite being large, this degradation is not statistically significant.

[11] We also see a slightly smaller improvement for the hand-crafted features using the modified French trees. The combination of tree kernels and hand-crafted features with the modified trees leads to a statistically significant improvement over the combination with the original trees.

Table 5.9: Tree kernel QE performance with modified French trees ($_m$: modified trees)

| | BLEU | | 1-TER | | Meteor | |
|---|---|---|---|---|---|---|
| | RSME | r | RSME | r | RSME | r |
| SyTK/C-T | 0.1608 | 0.1479 | 0.1925 | 0.2018 | 0.1620 | 0.2124 |
| SyTK/C-T$_m$ | **0.1591** | **0.2143** | 0.1940 | 0.1700 | **0.1602** | **0.2580** |
| SyTK/CD-ST | 0.1581 | 0.2437 | 0.1888 | 0.2774 | 0.1595 | 0.2715 |
| SyTK/CD-ST$_m$ | **0.1574** | **0.2609** | **0.1880** | **0.2918** | **0.1588** | **0.2862** |

Table 5.10: Tree kernel QE performance with modified French trees ($_m$: modified trees)

| | 1-HTER | | HBLEU | | Adequacy | | Fluency | |
|---|---|---|---|---|---|---|---|---|
| | RSME | r | RSME | r | RSME | r | RSME | r |
| SyTK/C-T | 0.2324 | 0.3068 | 0.2787 | 0.2982 | 0.2277 | 0.4127 | 0.2458 | 0.4821 |
| SyTK/C-T$_m$ | **0.2302** | **0.3331** | 0.2778 | 0.3114 | 0.2265 | 0.4236 | 0.2444 | 0.4912 |
| SyTK/CD-ST | 0.2267 | 0.3693 | 0.2721 | 0.3559 | 0.2258 | 0.4306 | 0.2431 | 0.5013 |
| SyTK/CD-ST$_m$ | **0.2257** | **0.3800** | 0.2715 | **0.3622** | 0.2253 | **0.4359** | 0.2425 | **0.5056** |

tic 1 responsible. The other 2 heuristics are helpful in this setting. Nevertheless, and ignoring the fact that this performance drop is not statistically significant, this suggests that the success of using modified French trees in improving tree kernel performance depends on the data set, score distribution, problem setting and even the task in hand, and may not be generalisable. We explore this question by applying the modification to QE with the SymForum data set as a different data and to *parser accuracy prediction* as a different task (and data) in the next two sections.

### 5.4.2 The SymForum Data Set

We investigate the effect of the French tree modification heuristics on the parse trees of the SymForum data set here. Table 5.10 compares the performance of the tree kernel QE systems using original and modified target side (French) trees.

When only target side constituency trees are used (SyTK/C-T$_m$), all scores improve using modified trees, but only the score differences of HTER prediction are statistically significant. Compared to the changes observed using the News data set,

Table 5.11: Parser Accuracy Prediction (PAP) performance with tree kernels using original and modified French trees ($_m$)

| | RSME | r |
|---|---|---|
| PAP | 0.1239 | 0.4035 |
| PAP$_m$ | 0.1233 | **0.4197** |

1) the improvements are smaller and 2) the modified trees are useful for HTER prediction, though not directly comparable to TER prediction in that case, confirming that the effectiveness of the modification depends on the data set and settings.

Interestingly, when the modified trees are used in the full tree kernel system SyTK/CD-ST$_m$, we see more statistically significant changes than when they are used alone in SyTK/C-T$_m$, in spite of the fact that three other types of trees are also involved in the full system; all Pearson r differences are statistically significant plus the RMSE difference of HTER prediction, despite their small size. This indicates a synergy between the modified French constituency trees and the other three types of trees, i.e. English constituency and dependency trees and French dependency trees.

### 5.4.3 Parser Accuracy Prediction

To further validate the effectiveness of the French tree modification heuristics, we choose a different task, parser accuracy prediction, the aim of which is to predict the accuracy of a parse tree without a reference (QE for parsing). The task was previously explored for English by (Ravi et al., 2008). We build a tree kernel model to predict the accuracy of French parses. To train the system, we parse the training section of FTB with our French parser and score them using $F_1$. We use the FTB development set to tune the SVM $C$ parameter and test the model on the FTB test set. Two parser accuracy prediction models are then built using this setting, one with the original parse trees and the second with the modified parse trees produced using the three heuristics listed above. The results are presented in Table 5.11. PAP is the system with original parse trees and PAP$_m$ with their modified version.

Both RMSE and Pearson r improve with the modified trees, where the r improvement is statistically significant. Although the improvement we observe is not as large as the one we observed for the QE for MT task, the results add weight to our claim that the structure of the FTB trees should be optimised for use in tree kernel learning.

## 5.5 Summary and Conclusion

We conducted an in-depth analysis of the syntax-based quality estimation based on the systems built in Chapter 4. We investigated the effect of parser accuracy on the performance of the syntax-based quality estimation systems by varying the accuracy of the parses used by these systems. The low accuracy parsers were generated by using only a fraction of the training data used by the original parsers, as well as using the output of a lower split/merge cycle of our iterative PCFG-LA parsers. We showed that parser accuracy measured by intrinsic metrics does not predict the accuracy of quality estimation built on these parses. This was confirmed on both data sets and using various parsers. This led to a rejection of the hypothesis put forth in the previous chapter that the noisy forum text parsing was the reason behind the syntax-based QE being less effective with SymForum data set. Instead, we conjecture that this can be attributed to a different aspect of parsing quality than that measured by classic PARSEVAL metric such as inconsistency of the parsing within the document due to structural inconsistencies of the forum text.

By teasing apart the roles played by the source and target syntax in QE, we observed that French constituency trees in the target side were far less useful than those of the English trees in the source side. Building the same QE systems in the opposite directing using the same data, we found that the poor quality parses of machine translation output resulting from its potentially ill-formed nature was not responsible for this performance gap. Instead, the structure of the French Treebank based on which our parsers were built found to be the culprit. We introduced a

set of heuristics which added more structure to French treebank. The resulting constituency trees proved to be more useful for QE systems using them especially via tree kernels and on the News data set.

# Chapter 6

# Semantic Role Labelling of French

One of the objectives of this thesis is to investigate the utility of semantic information in the quality estimation of machine translation. Such information, when extracted from the source and its translation, can help estimate how much of the meaning of the source is retained in the translation, i.e. translation adequacy. Semantic role labelling (SRL) (Gildea and Jurafsky, 2002) provides a shallow level of semantic analysis in the form of who did what to whom how, where and when, etc. in a sentence. Although SRL is not a complete representation of the semantics of the sentence, it can be useful in measuring the adequacy of the translation.

SRL is the task of identifying the predicates in a sentence, their semantic arguments along with the roles each of those argument takes. The outcome of the process is the predicate-argument structure of the sentence. Figure 6.1 shows an example of a sentence for which the predicate and its arguments are marked. The last decade has seen considerable attention paid to SRL (Màrquez et al., 2008), thanks to the existence of two major hand-crafted resources for English, namely FrameNet (Baker et al., 1998) and PropBank (Palmer et al., 2005). However, apart from English, only a few languages have SRL resources and these resources tend to be of limited size compared to the English ones. These languages include German, Japanese, Spanish, Catalan, Czech and Chinese, mostly introduced in the CoNLL-2009 shared task (Hajič et al., 2009) on syntactic and semantic dependencies in multiple languages.

[The index]$_{\texttt{Arg1}}$ [rose]$_{\texttt{predicate}}$ [1.1%]$_{\texttt{Arg2}}$ [last month]$_{\texttt{Arg-TMP}}$.

Figure 6.1: Semantic role labelling of the WSJ sentence: *The index rose 1.1% last month.*

French is one of those languages which suffer from a scarcity of hand-crafted SRL resources. The only available gold-standard resource is a small set of 1000 sentences from Europarl (Koehn, 2005) and manually labelled with PropBank verb predicates (van der Plas et al., 2010b). This dataset, however, is small and its main purpose is to evaluate various SRL approaches of projecting semantic role annotation of English sentences to their translations in French (van der Plas et al., 2011).

Semantic-based quality estimation approaches taken in this work are essentially based on the predicate-argument structure correspondence between the source text and its translation. Therefore, it is important that these structures are accurately extracted from both source and its translation. Consequently, if for example a predicate is identified by the semantic role labeller of the source text but its translation is missed by the semantic role labeller of the target text whilst it has correctly been translated, the quality estimation system will be misled. This requires not only the SRL of both source and target sides to be of high quality but also a balanced quality. However, the lack of French SRL resources can be an obstacle to the fulfilment of these requirements. Therefore the focus in this chapter is on addressing this problem.

We build on the work of van der Plas et al. (2010b) who tackle the lack of a reasonably-sized SRL data set for French, by building a large, "artificial" or automatically labelled dataset of approximately 1M Europarl sentences by projecting the semantic role labelling from English sentences to their French pairs and use it for training an SRL system. In particular, we attempt to answer the following questions:

- *How much artificial data is needed to train an SRL system?*
- *Is it better to use direct translations than indirect translations*, i.e. is it better to use for projection a source-target pair where the source represents the

original sentence and the target represents its direct translation as opposed to a source-target pair where the source and target are both translations of an original sentence in a third language?

- *Is it better to use coarse-grained syntactic information (in the form of universal part-of-speech tags and universal syntactic dependencies) than to use fine-grained syntactic information?*
- *What type of word alignments are more useful for projection?*
- *Is a large set of this artificial data better than a small set of hand-annotated data?*

The answers to these questions will help us find the best method to obtain the semantic role labels of our French data.

In the rest of this chapter, we review the work carried out on semantic role labelling of French in Section 6.1. We then introduce the data we use as well as the SRL system and evaluation setting in Section 6.2. Section 6.3 then presents the experiments.

## 6.1 Related Work

There have been relatively few works addressing semantic role labelling of French. While one research direction tries to avoid relying on hand-crafted data by unsupervised training of SRL models, the existing work mostly focuses on automatically or semi-automatically generating the resources for this language

Lorenzo and Cerisara (2012) propose a clustering approach for verb predicate and argument labelling (but not identification). They cluster verbs into 300 and 60 and argument roles into 40 and 10 clusters for English and French respectively. Since the approach is unsupervised, it can be applied to any language and labelling scheme. They choose VerbNet style roles (Schuler, 2006) and manually annotate a set of sentences with them for evaluation, achieving an $F_1$ of 78.5.

Gardent and Cerisara (2010) propose a method for semi-automatically annotating the French dependency treebank (Candito et al., 2010) with PropBank core roles (no adjuncts). They first manually augment TreeLex (Kupść and Abeillé, 2008), a syntactic lexicon of French, with semantic roles of syntactic arguments of verbs (i.e. verb subcategorization). They then project this annotation to verb instances in the dependency trees. They evaluate their approach by performing error analysis on a small sample and suggest directions for improvement. The annotation work is however at its preliminary stage and no data is published.

As mentioned earlier, van der Plas et al. (2011) use word alignments to project the semantic role labelling of the English side of EuroParl to its French side resulting in a large artificial dataset. This idea is based on the *Direct Semantic Transfer* hypothesis adapted from the *Direct Correspondence Assumption* hypothesis of Hwa et al. (2005) for syntactic dependency trees. It assumes that a semantic relationship between two words in a sentence can be transferred to any two words in the translation which are aligned to these source side words. Due to language variations and translation shifts, this assumption is strong and will not always hold. Therefore, they additionally filter the sentences with incomplete projections. According to the results, the projected annotations suffer mainly from recall, whereas the precision is acceptable. Evaluation on their 1K manually-annotated dataset shows that a joint syntactic-semantic dependency parser trained on this artificial data set performs significantly better than directly projecting the labelling from its English side. This is promising because in a real-world scenario, the English translations of the French data to be annotated do not necessarily exist.

Padó and Lapata (2009) also make use of word alignments to project semantic role labelling from English to German. The word alignments are used to compute the semantic similarity between syntactic constituents. In order to determine the extent of semantic correspondence between English and German, they manually annotate a set of parallel sentences and find that about 72% of the frames and 92% of the argument roles exist in both sides, ignoring their lexical correspondence.

They consider these numbers reassuring as such disagreements are expected even within a single language between the annotators. The projection is carried out at two levels: words and syntactic constituents. The best projection performance is achieved via constituent-based projection which is 56 $F_1$ points. Similar to the projection performance of van der Plas et al. (2011), recall is considerably lower than the precision. The results show that constituent-based projections are more robust to word alignment errors but further suffer from the SRL errors in the source and the parsing errors in the target.

## 6.2 Experimental Setup

### 6.2.1 Data

We use the two datasets described by van der Plas et al. (2011) and the delivery report of the *Classic* project (van der Plas et al., 2010a): the gold standard set of 1K sentences and the synthetic data set consisting of about 980K sentences.[1]

The gold standard data set (henceforth known as *Classic 1K*) was built by manually identifying each verb predicate, finding its equivalent English frameset in PropBank and identifying and labelling its arguments based on the description of the frameset. The synthetic dataset on the other hand was created as follows: the English side of an English-French parallel corpus (Europarl) was parsed using the joint syntactic-semantic parser described by Titov et al. (2009); the English and French sentences were then word-aligned using GIZA++ (Och and Ney, 2003). The dependency parses of the French side were produced using the `ISBN` parser (Titov and Henderson, 2007) described in Section 4.3 of Chapter 4 and the French SRLs by projecting the English SRLs via the word alignments. This dataset will be henceforth known as *Classic 980K*.

[1] We obtained the data by contacting the authors.

### 6.2.2 SRL

We use LTH (Björkelund et al., 2009), a dependency-based SRL system, in all of our experiments. This system was among the best-performing systems in the CoNLL-2009 shared task (Hajič et al., 2009) and is straightforward to use. It comes with a set of features tuned for each shared task language (English, German, Japanese, Spanish, Catalan, Czech, Chinese). Although French is not included among those languages, since the features are language independent,[2] we can borrow feature sets tuned for other languages.[3] We compared the performance of the English and Spanish feature sets on French and chose the former due to its higher performance (by 1 $F_1$ point).[4]

To evaluate SRL performance, we use the CoNLL-2009 shared task scoring script.[5] This scoring scheme assumes a semantic dependency between the argument and predicate and the predicate and a dummy root node. It then calculates the unlabelled and labelled precision (P), recall (r) and $F_1$ of these dependencies. We report unlabelled and labelled scores for each experimental setting.

## 6.3 Experiments

### 6.3.1 Learning Curve

The ultimate goal of SRL projection is to build a training set which partially compensates for the lack of hand-crafted resources. van der Plas et al. (2011) report encouraging results showing that training on their projected data is beneficial over directly obtaining the annotation via projection which is not always possible. Although the quality of such automatically generated training data may not be com-

[2]The system has a reranker which has a language-specific feature. We however do not use the reranker for labelling French

[3]The features for each language are selected from a larger repository of features, via a feature tuning procedure for those languages.

[4]Note that the Classic data is annotated for only verb predicates. Therefore, we remove features for nominal predicates from all LTH feature sets.

[5]`https://ufal.mff.cuni.cz/conll2009-st/eval09.pl`

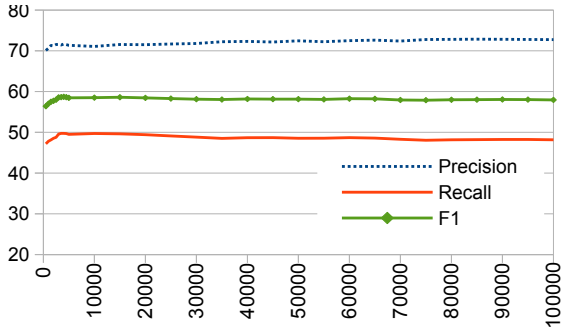



Figure 6.2: Learning curve with 100K training data of projected annotations

parable to the manual one, the possibility of building much bigger data sets may provide some advantage. Our first experiment investigates the extent to which the size of the synthetic training set can improve performance.

We randomly select 100K sentences from Classic 980K, shuffle it and split it into 20 subsets of 5K sentences. We then split the first 5K into 10 sets of 500 sentences. Finally, we train SRL models on the resulting 29 subsets using LTH. The performance of the models evaluated on the Classic 1K are presented in the learning curve of Figure 6.2. Surprisingly, the best $F_1$ (58.71) is achieved with only 4K sentences, and after that point the recall and subsequently $F_1$ tends to drop though precision shows a positive trend. This suggests that the additional sentences do not bring more information. The large gap between precision and recall is also interesting, showing that the complete spectrum of semantic roles is not covered by the projected data. This can be observed in Table 6.1 where the counts of ten most frequent arguments in the source side of the best-performing 5K sample, their counts in the target side (obtained by projection) as well as their ratios are presented. According to the figures, less than half of the `A1` arguments as the most frequent argument in the source side appear in the target side. This ratio is even smaller for most of other arguments, especially for `A2` as the third most frequent argument.

Table 6.1: The counts of ten most frequent arguments in the source and target side of a 5K projected sample and their ratios

| | Source # | Target # | Ratio |
|---|---|---|---|
| A1 | 13,845 | 6,209 | 44.8% |
| A0 | 8,917 | 5,609 | 62.9% |
| A2 | 3,375 | 751 | 22.3% |
| AM-TMP | 1,684 | 419 | 24.9% |
| AM-MOD | 1,666 | 767 | 46.0% |
| AM-MNR | 1,323 | 208 | 15.7% |
| AM-LOC | 1,094 | 367 | 33.5% |
| AM-ADV | 989 | 253 | 25.6% |
| AM-DIS | 689 | 405 | 58.8% |
| AM-NEG | 486 | 313 | 64.4% |

### 6.3.2 Direct Translations

Each sentence in the Europarl corpus was written in one of the official languages of the European Parliament and translated to all of the other languages. Therefore, both sides of a parallel sentence pair can be indirect translations of each other. van der Plas et al. (2011) suggest that translation divergence may affect automatic projection of semantic roles. They therefore select for their experiments only those 276K sentences from the 980K which are direct translations between English and French.[6]

Motivated by this idea, we replicate the learning curve presented in Section 6.3.1 with another set of 100K sentences randomly selected from only the direct translations. The curve is shown in Figure 6.3. There is no noticeable difference between this and the learning curves in Figure 6.2, suggesting that the projections obtained via direct translations are not of higher quality.

[6]The Classic data is only provided in 980K set and we did not have access to this 276K they have used. However, we utilized source language information available in the source release of Europarl to extract those direct translations.

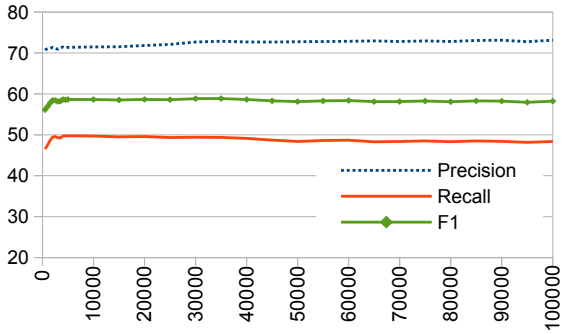



Figure 6.3: Learning curve with 100K training data of projected annotations on only direct translations

### 6.3.3 Impact of Syntactic Annotation

Being a dependency-based semantic role labeller, LTH employs a large set of features based on syntactic dependency structure. This inspires a comparison of the impact of different types of syntactic annotations on the performance of this system.

Based on the observations from the learning curves generated in previous sections, we choose two different sizes of training sets. The first set contains the first 5K sentences from the original 100K, as we saw that more than this amount tends to diminish performance. The second set contains the first 50K from the original 100K, the purpose of which is to check if changing the parses affects the usefulness of adding more data. We will call these data sets *Classic 5K* and *Classic 50K* respectively. The following sections describe each syntactic annotation and present the performance of SRL systems trained on the two data sets annotated with them.

#### 6.3.3.1 Universal POS Tags

Petrov et al. (2012) create a set of 12 universal part-of-speech (POS) tags which should in theory be applicable to any natural language. It is interesting to know whether these POS tags are more useful for semantic role labelling than the original set of the 29 more fine-grained POS tags used in the French Treebank which we have used so far. To this end, we convert the original POS tags of the data to universal POS tags and retrain and evaluate the SRL models. The results are given in Table 6.2 (`OrgDep+UniPOS`). The first row of the table (`Original`) shows the

Table 6.2: SRL performance using different syntactic parses with Classic 5K and 50K training sets

| | Classic 5K | | | | | |
|---|---|---|---|---|---|---|
| | Unlabelled | | | Labelled | | |
| | P | R | $F_1$ | P | R | $F_1$ |
| `Original` | 85.95 | 59.64 | 70.42 | **71.34** | 49.50 | **58.45** |
| `OrgDep+UniPOS` | **86.71** | **60.46** | **71.24** | 71.11 | **49.58** | 58.43 |
| `StdUniDep+UniPOS` | 86.14 | 59.76 | 70.57 | 70.60 | 48.98 | 57.84 |
| `CHUniDep+UniPOS` | 85.98 | 59.21 | 70.13 | 70.66 | 48.66 | 57.63 |
| | Classic 50K | | | | | |
| | Unlabelled | | | Labelled | | |
| | P | R | $F_1$ | P | R | $F_1$ |
| `Original` | 86.67 | 58.07 | 69.54 | **72.44** | 48.54 | 58.13 |
| `OrgDep+UniPOS` | **86.82** | 58.71 | **70.05** | 72.30 | **48.90** | **58.34** |
| `StdUniDep+UniPOS` | 86.38 | **58.90** | 70.04 | 71.61 | 48.83 | 58.07 |
| `CHUniDep+UniPOS` | 86.47 | 58.26 | 69.61 | 71.74 | 48.34 | 57.76 |

performance using the original annotation. Even though the scores increase in most cases compared to those of the `Original` setting – due mostly to a rise in recall – the changes are small. It is worth noting that the identification task (unlabelled scores) seems to benefit more from the universal POS tags.

#### 6.3.3.2 Universal Dependencies and POS Tags

Similar to universal POS tags, McDonald et al. (2013) introduce a set of 40 universal dependency types which generalize over the dependency structure specific to several languages. For French, they provide a new treebank, called *uni-dep-tb*, manually annotating 16,422 sentences from various domains with these dependencies. We now explore the utility of this new dependency scheme in semantic role labelling.

The French universal dependency treebank comes in two versions. The first version uses the standard dependency structure based on basic Stanford dependencies (de Marneffe and Manning, 2008) where content words are the heads except in copula and adposition constructions. The second version treats content words as the heads for all constructions without exemption. We use both schemes in order to verify their effect on SRL.

In order to obtain universal dependencies for our data, we train parsing models with MaltParser (Nivre et al., 2006) using the entire uni-dep-tb.[7] There are two types of POS tag sets in the French uni-dep-tb. One is coarse-grained containing the universal POS tags described earlier and the other is fine-grained containing an additional two tags to the universal ones: `AUX` for auxiliary and `PNOUN` for pronouns. Since we do not have a tagger to obtain these fine tags for our data, we only experiment with the universal POS tags.[8] We then parse our data using these MaltParser models. The input POS tags to the parser are the universal POS tags used in `OrgDep+UniPOS`. We train and evaluate new SRL models on this data. The results are shown in Table 6.2. `StdUniDept+UniPOS` is the setting using standard dependencies and `CHUDep+UPOS` using content-head dependencies. In comparing the performance between when the original parses and when the universal dependency parses are used, it should be noted that, in addition to the difference in dependency schemes, the parsers as well as their training data are different.

Since we are using the same universal POS tags in training SRL models, the effect of universal dependencies can directly be compared to those of the original ones by comparing these results to `OrgDep+UniPOS`. According to the results, the use of universal dependencies has only a modest (negative) effect. It however appears that content-head dependencies are slightly less useful than standard dependencies.

Overall, we observe that the universal annotations can be reliably used when the fine-grained annotation is not available. This can be especially useful for languages which lack such resources and require techniques such as cross-lingual transfer to replace them.

[7]Based on our preliminary experiments on the parsing performance, we use `LIBSVM` as learning algorithm, `nivreeager` as the parsing algorithm for the standard dependency models and `stackproj` for the content-head ones.

[8]This is possible either by replacing fine-grained tags with universal ones in the data or changing the MaltParser feature files to make use of the `CPOSTAG` column instead of the `POSTAG` one.

Table 6.3: Projecting English SRL from source side of Classic 1K data to the target side using various alignments

| | Unlabelled | | | Labelled | | |
|---|---|---|---|---|---|---|
| | P | R | $F_1$ | P | R | $F_1$ |
| `Intersection` | **79.32** | 25.45 | 38.53 | **56.76** | 18.21 | 27.57 |
| `Source-to-target` | 71.17 | 42.58 | **53.29** | 47.33 | 28.32 | **35.43** |
| `Union` | 55.76 | **50.56** | 53.04 | 36.04 | **32.68** | 34.28 |

### 6.3.4 The Impact of Word Alignment

We observed that projection via the intersection of word alignments in the two translation directions was too constrained so that a significant amount of labelling could not be transferred to the target. We build a system with intersection alignments as the baseline and compare it to the projections via two other alignment heuristics: source-to-target and union. The source-to-target method loosens the restriction of the intersection method by selecting all the alignments in this direction no matter whether they match those in the opposite direction. The union method lifts all the restrictions and merges the alignments of both directions.

We evaluate the projections using these word alignments on the Classic 1K data set. The source side is labelled with LTH trained on the whole CoNLL-2009 shared task data set and the alignments are obtained using GIZA++ via Moses toolkit (Hoang et al., 2009). The results are shown in Table 6.3.

As expected, union alignments compensate for the very low recall of intersection ones, but at the cost of precision. However, the cost is not as much as the gain so that the resulting $F_1$ is considerably higher with union alignments. On the other hand, source-to-target alignments lie in between in terms of precision and recall. Nevertheless, its $F_1$ scores are the highest with a slight difference to those of union alignments.

Overall, both of the new alignments seem to suit projection better than the intersection ones. This is in contrast to other projection works (Padó and Lapata, 2009; van der Plas et al., 2011) which have used intersection alignments with the hope

Table 6.4: Training French SRL on projected English SRL from source side of Classic 5K data to the target side using various alignments (`Intsc`: intersection, `S2T`: source-to-target)

| | Unlabelled | | | Labelled | | |
|---|---|---|---|---|---|---|
| | P | R | $F_1$ | P | R | $F_1$ |
| `Intersection` | **84.54** | 43.69 | 57.61 | **69.76** | 36.05 | 47.53 |
| `Source-to-target` | 82.92 | 51.76 | 63.73 | 67.07 | 41.86 | 51.55 |
| `Union` | 82.09 | **59.49** | **68.99** | 64.19 | **46.52** | **53.95** |

that less noisier alignments will lead to higher projection performance.[9] Between union and source-to-target alignments, one can choose based on the requirement of the downstream task in terms of the priority of precision or recall. Union alignments seem to be more balanced in this respect.

In addition to evaluating the effect of word alignment algorithms on directly projecting the annotation to the target data, we also evaluate the performance of a SRL model trained on the projected annotations and applied on this target data. We project the original SRL annotation of the source side of the Classic 5K to its target side using the three alignment heuristics we used above. We then train an SRL model using each of these three projected annotations and evaluate it on the Classic 1K data set. The performances are given in Table 6.4.[10]

Expectedly, the model trained on the projections of intersection alignments achieves a higher precision. However, the precision gaps between the three methods is not as much as those seen in the direct projection results in Table 6.3. This probably occurs because some of the erroneous labels resulting from noisy alignments are not given a high importance during the training of the SRL system due to their inconsistency.

Recall results also follow the same pattern as in direct projection; lifting the restrictions increases recall. The recall gaps are bigger than the precision gaps so

[9]We tried all the other alignment heuristics implemented in Moses. They all ranked higher than the intersection but lower than the union and source-to-target alignments.

[10]Note that the 5K model performance using intersection alignments for projection in this table is not comparable to the 5K model built on the original projection presented in Table 6.2 (first row) which also used intersection alignments because we were not able to replicate their projections.

that the resulting $F_1$ scores are significantly higher with the new word alignments. However, they are not as big as those we observed in direct projection results.

One difference with direct projection is that union alignments now lead to the highest $F_1$ rather than source-to-target alignments. Again, the choice of alignments can be based on the priority of precision or recall in the downstream application. When there is no difference, union alignments seem to be the best option.

Another notable difference is that the performance of the semantic role labelling using a model trained on projected annotations is substantially higher than directly projecting them. This is in par with van der Plas et al. (2011) results.

### 6.3.5 Quality vs. Quantity

In Section 6.3.1, we saw that adding more data annotated through projection did not improve the performance of semantic role labelling. In other words the same performance was achieved using only a small amount of data. This is contrary to the motivation for creating synthetic training data, especially when the hand-annotated data already exist, albeit in a small size. In this section, we compare the performance of SRL models trained using manually annotated data with SRL models trained using 5K of artificial or synthetic training data. We use the original syntactic annotations for both data sets.

To this end, we carry out a 5-fold cross-validation on the Classic 1K. We then evaluate the Classic 5K model, on each of the 5 test sets generated in the cross-validation. The average scores of the two evaluation setups are compared. The results are shown in Table 6.5.

While the `5K` model achieves higher precision, its recall is far lower resulting in dramatically lower $F_1$. This high precision and low recall can be attributed to the low confidence of the model trained on projected data due to a considerable amount of information not transferred during the projection. A possible reason can be that the Classic projection uses intersection of alignments in the two translation directions, which is the most restrictive setting and leaves many source predicates

Table 6.5: Average scores of 5-fold cross-validation with Classic 1K, 5K, 1K plus 5K and self-training with 1K seed and 5K unlabelled data (`SelfT`)

| | Unlabelled | | | Labelled | | |
|---|---|---|---|---|---|---|
| | P | R | $F_1$ | P | R | $F_1$ |
| `1K` | 83.76 | 83.00 | 83.37 | 68.40 | 67.78 | **68.09** |
| `5K` | **85.94** | 59.62 | 70.39 | 71.30 | 49.47 | 58.40 |
| `1K+5K` | 85.74 | 66.53 | 74.92 | **71.48** | 55.46 | 62.46 |
| `SelfT` | 83.82 | **83.66** | **83.73** | 67.91 | **67.79** | 67.85 |

and arguments unaligned, as seen in the previous section.

In addition to comparing the performance of the two data sets, we verify the effect of utilizing both data sets at the same time in two different ways. First, we add the Classic 5K data set to the training section of Classic 1K data in each fold of another cross-validation setting and evaluate the resulting models on the same test sets. The results are reported in the third row of the Table 6.5 (`1K+5K`). As can be seen, the low quality of the projected data significantly degrades the performance compared to when only manually annotated data is used for training.

Second, based on the observation that the quality of labelling using manually annotated data is higher than using the automatically projected data, we replicate `1K+5K` with the 5K data labelled using the model trained on the training subset of `1K` at each cross-validation fold. In other words, we perform a one-round self-training with this model. The performance of the resulting model evaluated in the same cross-validation setting is given in the last row of Table 6.5 (`SelfT`).

As expected, the labelling obtained by models trained on manual annotation is more useful for training than the labelling obtained by the projected ones. It is worth noting that, unlike with the `1K+5K` setting, the balance between precision and recall follows that of the `1K` model. In addition, some of the scores are the highest among all results, although the differences are not big.

Table 6.6: Average scores of 5-fold cross-validation with Classic 1K and 5K using 200 sentences for training and 800 for testing at each fold

| | Unlabelled | | | Labelled | | |
|---|---|---|---|---|---|---|
| | P | R | $F_1$ | P | R | $F_1$ |
| 1K | 82.34 | **79.61** | **80.95** | 64.14 | **62.01** | **63.06** |
| 5K | **85.95** | 59.64 | 70.42 | **71.34** | 49.50 | 58.45 |

### 6.3.6 How little is too little?

In the previous section we saw that using manually annotated data as small as 800 sentences resulted in significantly better SRL performance than using projected annotation as large as 5K sentences. This unfortunately indicates the need for human labour in creating such resources. It is interesting however to know the lower bound of this requirement. To this end, we reverse our cross-validation setting and train on 200 and test on 800 sentences. We then compare to the 5K models evaluated on the same 800 sentence sets at each fold. The results are presented in Table 6.6.

According to the results, even with only 200 manually annotated sentence, the performance is considerably higher than with 5K sentences of the projected annotations. However, as one might expect, compared to when 800 sentences are used for training, this small model performs significantly lower.

## 6.4 Summary and Conclusion

.

We explored the projection-based approach to semantic role labelling by carrying out experiments with a large set of French sentences annotated automatically by transferring the labelling from English. We found that increasing the number of these artificial projections that are used in training an SRL system does not improve performance as might have been expected when creating such a resource. Instead it is better to train directly on what little gold standard data is available, even if this dataset contains only 200 sentences. Using a 5-fold cross-validation on the dataset of

1K gold standard annotations, the SRL performance was substantially higher (>10 $F_1$ points in terms of both unlabelled and labelled scores) than when a much larger set of projected annotations is used for training.

Moreover, the experiments showed that less restrictive alignment extraction strategies including extracting the union of the two sets or only source-to-target alignments lead to a better recall and consequently $F_1$ both when used for direct projection to the test data or indirectly for creating the training data and then applying the model on the test data. The union alignments result in a lower precision due to introducing more noise but in a higher recall, whereas the source-to-target alignments leads to a higher precision but a lower recall. The resulting $F_1$ however is around the same for both approaches.

We also compared the use of the universal part-of-speech tags and dependencies to the original, more fine-grained sets and showed that they can be used in SRL with only a little difference in the performance.

Based on the observations from the experiments in this chapter, the best model for semantic role labelling of the French translations is the one trained on the small available hand-annotated data set of 1000 sentences. We will use such a model in semantic role labelling of the French side of our data set for semantic-based quality estimation in the next chapter.

# Chapter 7

# Semantic-based Quality Estimation

An important criterion in assessing the quality of translation is its adequacy. While the fluency of translation concerns its syntax, the adequacy of translation is related to its semantics.[1] In general, for a translation to be adequate, its semantic analysis should comply with that of its source. Therefore, to automatically judge the adequacy of a translation, both the source and the translation must be semantically analysed. The comparison of these analyses will expose the extent to which the meaning of the source is retained in the translation. Although this procedure may appear to be simple, there are certain challenges hindering the process as all of these steps should be taken automatically. Representing the meaning of two sentences each in a different language in a unified framework is never an easy task. Moreover, the state-of-the-art automatic semantic analysers are far from perfect. This problem is further exacerbated when they are applied to user-generated content and the machine translation output.

As explained in the previous chapter, semantic role labelling is a well established shallow semantic representation framework, in which a sentence is decomposed into

[1]It is worth noting that the notion of adequacy used here is the one adopted by the SMT community (e.g. NIST, Euromatrix available at `http://www.euromatrix.net/deliverables/Euromatrix_D1.3_Revised.pdf`), while the translation studies scholars such as Al-Qinai (2000) use the term adequacy in a more generic manner.

its propositions each represented by a predicate and its arguments, i.e. predicate-argument structure. Although SRL does not deeply represent the meaning of a sentence, its source/target correspondence can provide useful information about the adequacy of the translation. The previous chapter described the method we use to acquire semantic role labelling for French. Besides, semantic role labelling of English is well studied and reasonable resources are available for it. In this chapter, we attempt to utilize this information in estimating the quality of machine translation. We use the same methods as those we used for syntax-based QE in Chapter 4, i.e. tree kernels and hand-crafted features. We additionally design a QE metric which is directly based on the *Predicate-Argument Structure Match* (*PAM*) between the source and its translation. The metric is used in two ways: 1) as a measure of translation quality by itself, 2) as an indicator of translation quality incorporated as a feature into our hand-crafted QE system. The ultimate outcome of the experiments in this chapter is the QE system built by combining the syntax-based QE system of Chapter 4 with the semantic-based QE system presented in this chapter. This system is also combined with the baseline features introduced in 4.2.1.

In the rest of this chapter, we first discuss the related work in using semantics in QE in Section 7.1. In Section 7.2, we describe the data and its semantic role labelling. We then move on to the experiments which start with a the tree kernel approach to semantic-based QE (Section 7.3) followed by the hand-crafted approach (Section 7.4). Next, we introduce PAM, our QE metric, and various ways of using it in quality estimation. Finally, we combine the QE systems we have developed in this work.[2]

## 7.1 Related Work

Pighin and Màrquez (2011) propose a sophisticated method for ranking two translation hypotheses that exploits the projection of SRL from source to its translation

[2]For easy comparison, all the experimental results in this chapter are also presented in Table A.1 in Appendix A beside other QE results.

using word alignments produced by a constrained translation. They first project SRL of a source corpus to its parallel reference corpus and then build two translation models using it: 1) translations of proposition labelling sequences in the source to its projection in the target (e.g. `A0 verb A1` to `A0 A1 verb`) and 2) translations of argument role fillers in the source to their projected counterparts in the target. They then project source SRL to its machine translation and force the above models to translate source proposition labelling sequences to the projected ones. They finally use the confidence scores of these translations and their reachability (whether the forced translation was possible) to train a classifier which selects the better of the two translation hypotheses. The constrained model they use to generate the alignments fails to produce the translation in 35% of the cases. They highlight this problem as the main limitation of their method. They additionally blame the low recall of the SRL: 6% of the sentences could not be labelled with the SRL due to the lack of verb and 3% of the labellings could not be projected due to the loss of predicate during translation. The highest accuracy achieved by their binary classifier is 64%.

Semantic roles have also been used in MT evaluation where reference translations are available. Giménez and Màrquez (2007) use semantic roles in building several MT evaluation metrics which seek to compensate for the shortcomings of other popular metrics such as BLEU in accounting for linguistic aspects of the translation. These metrics measure the full or partial lexical match between the fillers of same semantic roles in the hypothesis and translation, or simply the role label matches between them. Their results show that such metrics can be more useful in ranking the translations of heterogeneous systems. However, they conclude that these linguistic features cannot serve as a global measure of translation quality but only in combination with other features and metrics reflecting different aspects of the quality.

Lo and Wu (2011) introduce HMEANT, a manual MT evaluation metric based on predicate-argument structure matching which involves two steps of human en-

gagement: 1) semantic role annotation of the reference and machine translation, 2) evaluating the translation of predicates and arguments. The metric calculates the $F_1$ score of the semantic frame match between the reference and machine translation based on this evaluation. To keep the costs reasonable, the first step is done by amateur annotators who were minimally trained for only minutes with a simplified list of 10 thematic roles. On a set of 40 examples, they meta-evaluate the metric in terms of correlation with human judgements of translation adequacy ranking. They report as high a correlation as that of HTER.

Lo et al. (2012) propose MEANT, a variant of HMEANT, which automatizes its manual steps using 1) automatic SRL systems for (only) verb predicates and 2) automatic alignment of predicates and their arguments in the reference and machine translation based on their lexical similarity. Once the predicates and arguments are aligned, their similarities are measured using a variety of methods such as cosine and even Meteor and BLEU metrics. When the final score is computed, the similarity scores replace the counts of correct and partial translations used in HMEANT. This metric outperforms several automatic metrics including BLEU and Meteor and TER, but it significantly under performs HMEANT and HTER. Their investigations show that automatizing the second step does not affect the performance of MEANT. Therefore, it seems to be the lower accuracy of the semantic role labelling that is responsible for the performance gap with HMEANT.

When computing the HMEANT and MEANT scores, matching predicate and semantic roles are each multiplied by a weight which determines the contribution of each to the meaning of the sentence and consequently to the computation of the score. These weights are set by tuning them on a development set using a grid search. Lo and Wu (2013) introduce UMEANT, a variation of MEANT, which uses an unsupervised way of estimating these weights; the weight of each semantic role is its relative frequency in the reference translation set. They report that UMEANT achieves a competitive performance to MEANT.

Originally applied to Chinese-English translations, Bojar and Wu (2012) experi-

ment with HMEANT on English-Czech. They use the metric for ranking and select 50 sentences each with 13 machine translations for their experiments. They identify a set of flaws of the metric and propose solutions for them. The most important problems concern the semantic role annotation step and stem from the superficial SRL annotation guidelines. These problems can be exacerbated in MEANT due to the automatic nature of the two steps. In addition, when there is no verb predicate in the sentence, such as nominal construction (e.g. titles) or missing verbs as a result of erroneous translation, the sentence is ignored. Another problem they identify is simple role label set of HMEANT which falls short when annotating cases such as passive constructions or secondary objects. They suggest a metric variant where role labels are not taken into account.

More recently, Lo et al. (2014) extend MEANT to ranking translations without a reference, i.e. quality estimation. This metric is called XMEANT and uses 1) phrase translation probabilities for aligning semantic role fillers of the source and its translation and 2) bracketing ITG (Inversion Transduction Grammars) constraints on the word alignment of semantic role fillers. They claim that XMEANT outperforms MEANT for two reasons. First, the machine translation output is closer to the source sentence in terms of semantic structure than a reference translation is. The second reason concerns the new method of word-aligning the tokens inside semantic arguments which uses bracketing ITG constraints instead of the bag-of-words approach used in MEANT.

In our work, unlike both QE works introduced above, we estimate quality scores instead of ranking two translations. Our semantic-based approaches to QE work in both directions, i.e. we use statistical methods like Pighin and Màrquez (2011) to predict the quality and we design a metric for measuring the quality like Lo et al. (2014). This metric is simpler to compute and has no parameters involved for tuning. It has both labelled and unlabelled versions in which semantic role labels are ignored. Moreover, we combine various methods.

Table 7.1: CoNLL-2009 data used for training English SRL

| Data set | Size (#sentences) |
|---|---|
| Training set | 39,279 |
| Development set | 1,334 |
| WSJ test set | 2,399 |
| Brown test set | 425 |
| Sum | 43,437 |

## 7.2 Data and Annotation

The experiments are carried out using the SymForum data set introduced in Section 3.2.1 of Chapter 3. For semantic role labelling of both English and French data, we use LTH (Björkelund et al., 2009) as described in Section 6.2 of Chapter 6. For English, we train this system using all the data provided in the CoNLL-2009 shared task (Hajič et al., 2009) which includes the PropBank I training, development and test sets. The test sets include both WSJ and Brown test sets. Note that the training and development sets also come from WSJ corpus, so the Brown test set is from a different domain than the rest of the corpus. The sizes of the data sets are shown in Table 7.1. We use the reranking option of the tool which perform slightly better than the original setting. For French, based on our findings in Chapter 6, we train it on the Classic 1k manually annotated data. Since the French data set only annotates verb predicates, we only use the verb predicates of the English side as well, though the CoNLL-2009 shared task provides both verb and nominal predicate annotation. Since the reranking option of the LTH has a language-dependent feature, we do not use it for this data.

The syntactic infrastructure of the semantic role labelling of both English and French sides of the data set is the one introduced and used for syntax-based QE experiments in Section 4.3 of Chapter 4. The English SRL data comes with gold standard syntactic annotation. On the other hand, for our QE data set, such annotation is not available. Our preliminary experiments show that, since the SRL system heavily relies on syntactic features, the performance considerably drops when

Table 7.2: Performances of the English SRL system on various data sets and the French SRL system using 5-fold cross-validation

| | | Unlabelled | | | Labelled | | |
|---|---|---|---|---|---|---|---|
| | | P | R | $F_1$ | P | R | $F_1$ |
| English | WSJ test | 85.76 | 79.96 | 82.76 | 80.59 | 75.14 | 77.77 |
| | Brown test | 81.78 | 76.52 | 79.07 | 69.33 | 64.87 | 67.02 |
| French | Classic 1K | 83.69 | 81.62 | 82.62 | 68.54 | 66.84 | 67.66 |

the syntactic annotation of the test data is obtained using a different parser than that of the training data. We therefore replace the parses of the training data with those obtained automatically by first parsing the data using the Lorg parser and then converting them to dependencies using `Stanford` converter. The POS tags are also replaced with those output by the parser. For the same reason, we replace the original dependency parses of the French training data (Classic 1K) with those generated by the ISBN parser and its POS tagging with those output by the MElt tagger.[3] Table 7.2 shows the performances of our English and French SRL systems.

When trained only on the training section of PropBank, the English SRL achieves 77.77 and 67.02 labelled $F_1$ points on the WSJ and Brown test sets respectively. The French SRL evaluated with a 5-fold cross-validation on the Classic 1K data set obtains an $F_1$ average of 67.66. The large gap between unlabelled and labelled performance of the French SRL is noticeable, suggesting that its main problem is in assigning labels to the arguments it finds. The unlabelled performance is almost the same as that of the English SRL, evaluated on the in-domain WSJ test set, despite their incomparable training data size. More interestingly, the recall of the French SRL is even higher. This can also be seen in the number of predicates and arguments each of these systems find in our QE data set, where there is little difference between them. These numbers are given in Table 7.3.

However, the labelling (classification) performance of the French SRL is substan-

[3]The original dependency annotation of `Classic 1K` is also generated using ISBN parser. However, we found a slight difference with our replicated parses which was probably due to the size of training set and/or version difference.

Table 7.3: Number of predicates and arguments extracted from the SymForum data set by English and French SRL systems

| | #predicates | #arguments |
|---|---|---|
| Source (English) | 9133 | 21,393 |
| MT output (French) | 8795 | 20,024 |
| Post-edit (French) | 8875 | 21,134 |

tially lower compared to the English one, especially in terms of precision. This can lead to a quality imbalance between the predicate-argument structures extracted from the source and target sides of our QE data set which can subsequently interfere with the genuine imbalance caused by the quality of translation – which is in fact what we are trying to measure. Our analysis of the experimental results in the rest of this chapter will shed more light on this matter.

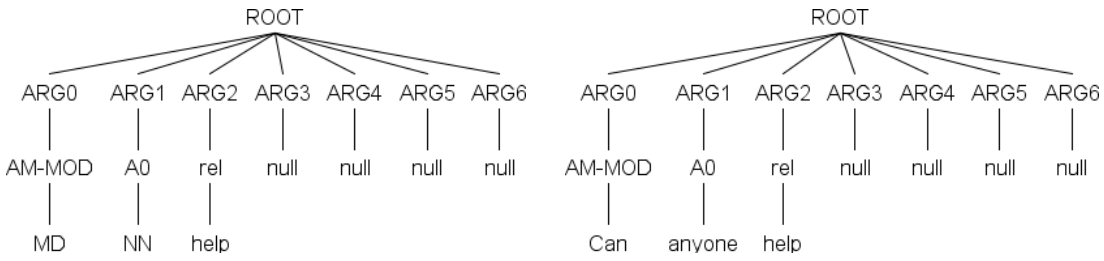
## 7.3 Semantic-based QE with Tree Kernels

In Section 4.2.2 of Chapter 4, we successfully used tree kernels in employing syntactic trees in quality estimation. In this section we use them to apply semantic role labelling encoded in trees in semantic-based QE. Similar to the syntax-based approach, we use both dependency-based and constituency-based semantic trees. The kernels in all semantic-based QE experiments are computed in the same way for the syntax-based QE systems as described in Section 4.2.2 of Chapter 4.

### 7.3.1 Dependency Tree Kernels

Unlike syntactic parsing, semantic role labelling does not produce a tree to be directly used in the tree kernel framework. However, there are various ways to convert the output of an SRL system to a tree. We explore three different methods as follows:

- `PAS` (*predicate-argument structure*) format introduced by Moschitti et al. (2006)
- `PST` (*proposition subtree*) introduced here
- `SAS` (*semantic-augmented syntactic trees*) based on syntactic tree kernels




(a) `PAS with POS tags` (b) `PAS with word forms`

Figure 7.1: Two variations of dependency `PAS` formats for the sentence: *Can anyone help?*

In the following sections, we describe each method with their variations and present the experimental results for each of them.

#### 7.3.1.1 PAS Format

In this format, a fixed number of nodes are gathered under a dummy root node as slots of one predicate and 6 arguments of a proposition[4] (one tree per predicate). Each node dominates an argument label or a dummy label for the predicate (`rel`), which in turn dominates the POS tag of the argument or the predicate lemma. If a proposition has more than 6 arguments they are ignored; if it has fewer than 6 arguments, the extra slots are attached to a dummy null label. Figure 7.1a shows an example tree in this format: the first slot is filled by `AM-MOD` which is the PropBank label for the modal adjunct role of *Can*, *rel* which is the label used for the predicate corresponding to the verb *help* and `A0` which is the PropBank label representing the agent role of *anyone*.

We build a tree kernel system using such PAS trees extracted from the predicate argument structures of the source and target. There will be one tree per proposition, thus the number of trees vary for each instance. The performance of this system (`SeTK/D-PAS`$_{POS}$) is shown in Table 7.4. The performance of the `B-WMT17` baseline built in Section 4.3.1 of Chapter 4 is repeated in the table (in grey) for comparison.

The PAS tree kernels perform statistically significantly lower than the baseline. This is somehow expected as there is very little structure encoded in the trees.

[4]We use proposition to refer to the predicate and its arguments.

Table 7.4: Dependency tree kernel systems

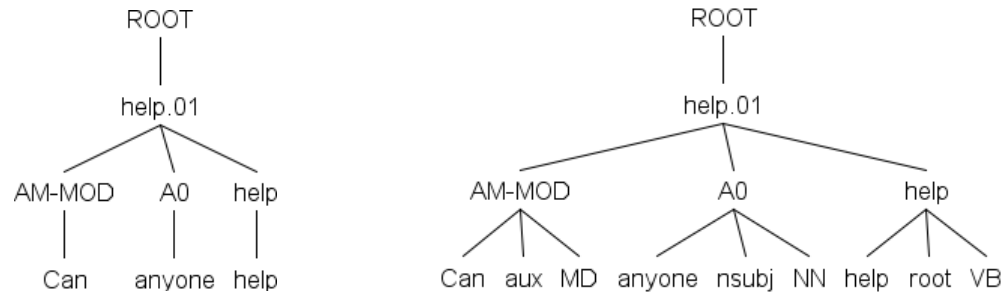
| | 1-HTER | | HBLEU | | Adequacy | | Fluency | |
|---|---|---|---|---|---|---|---|---|
| | RSME | r | RSME | r | RSME | r | RSME | r |
| B-WMT17 | 0.2310 | 0.3661 | 0.2696 | 0.3806 | 0.2219 | 0.4710 | 0.2469 | 0.4769 |
| SeTK/D-PAS$_{POS}$ | 0.2489 | 0.1774 | 0.2856 | 0.1843 | 0.2423 | 0.2770 | 0.2652 | 0.3252 |
| SeTK/D-PAS$_{word}$ | 0.2480 | 0.1789 | 0.2926 | 0.1669 | 0.2485 | 0.2660 | 0.2654 | 0.3221 |
| SeTK/D-PST$_{w}$ | 0.2413 | 0.2082 | 0.2832 | 0.2237 | 0.2431 | 0.2567 | 0.2663 | 0.3080 |
| SeTK/D-PST$_{wdp}$ | 0.2409 | 0.2136 | 0.2815 | 0.2450 | 0.2383 | 0.3169 | 0.2606 | 0.3670 |
| SeTK/D-SAS$_{afx}$ | 0.2270 | 0.3699 | 0.2738 | 0.3391 | 0.2291 | 0.4022 | 0.2476 | 0.4731 |
| SeTK/D-SAS$_{node}$ | 0.2271 | 0.3667 | 0.2727 | 0.3483 | 0.2275 | 0.4169 | 0.2443 | 0.4930 |
| SyTK/D-ST | 0.2261 | 0.3778 | 0.2722 | 0.3546 | 0.2280 | 0.4118 | 0.2455 | 0.4860 |

Additionally, this format does not make any use of the surface form of the source or translation sentences. We therefore build another system using word forms instead of part-of-speech tags in the leaves. Surprisingly, this system (SeTK/D-PAS$_{word}$) performs even worse than when POS tags are used as shown in Table 7.4 (except in predicting HTER), rendering the PAS format insufficient for this purpose.

#### 7.3.1.2 PST Format

We propose another format in which all *proposition subtrees* (PST) of the sentence are gathered under a dummy root node. A PST is formed by the predicate label (frameset) as the root, dominating its own lemma and all its arguments role labels. The lemma in turn dominates the word form of the predicate, and the argument roles dominate the word forms of their role fillers. Figure 7.2a shows an example PST of a sentence with only one proposition. The performance of the system built with trees in this format, (SeTK/D-PST$_{w}$), is presented in Table 7.4. It is similar to the performance of the PAS systems and far lower than the baseline. While human-targeted metric prediction improves, manual metric prediction degrades, when compared to the PAS systems.

We now add more information to the PST trees by adding the POS tag and dependency relation of the argument node to its head as siblings of the word form as shown in Figure 7.2b. The new system is named SeTK/D-PST$_{wdp}$ in Table 7.4.




(a) PST with word forms (b) PST with word forms, dependency labels and POS tags

Figure 7.2: Two Variations of dependency PST formats for the sentence: *Can anyone help?*

The new format achieves the highest results among the semantic tree kernel systems. The improvements are substantial in the case of manual metric prediction. However, the score are still far below the baseline.

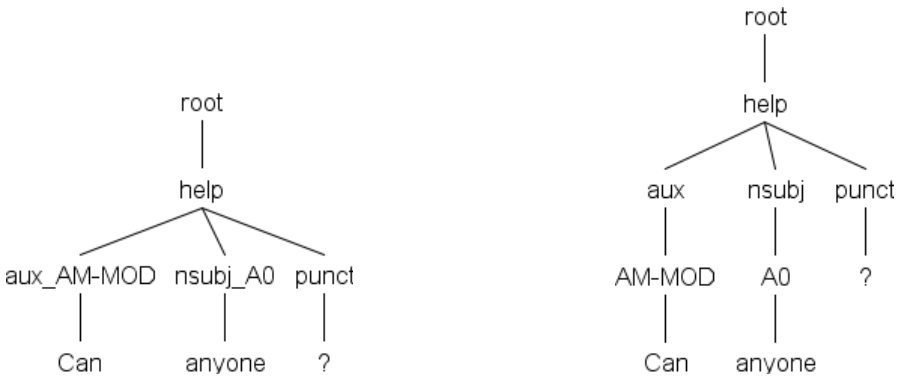
#### 7.3.1.3 SAS Format

The above formats motivated by the predicate-argument structure do not seem to capture enough information about the quality of translation. On the other hand, our experiments using syntactic tree kernels for this purpose have shown promise. Inspired by this, we turn our attention to augmenting syntactic tree kernels with semantic information instead of building proposition-based tree kernels.

We augment the dependency tree kernels introduced in Section 4.2.2 of Chapter 4 in two ways. The first method affixes the argument role label to the syntactic dependency label of the argument node. Figure 7.3a shows an example tree in this format. Using augmented trees of both source and target side, we build SeTK/D-SAS$_{afx}$. The performance of this system is shown in Table 7.4.

As the results show, this format clearly suits quality estimation better than the previous ones. It even outperforms the baseline in predicting HTER and performs close to the baseline in Fluency prediction. The difference with the baseline on HTER prediction is small in terms of Pearson r but considerable in terms of RMSE.

The second method differs from the first one in that it attaches the argument role label as a node under the syntactic dependency label. It then dominates the




(a) Argument labels as suffixes (b) Argument labels as nodes

Figure 7.3: Two Variations of dependency SAS formats for the sentence: *Can anyone help?*

argument itself. An example is given in Figure 7.3b. The system built with the augmented trees of the source and target in this way is named SeTK/D-SAS$_{node}$ in Table 7.4. As can be seen, integrating the argument role labels as nodes tends to be more useful than affixing them to the dependency relation nodes, perhaps due to a higher data sparsity level caused by the latter. The only slightly degraded score is the Pearson r of HTER prediction. The gaps are bigger for the case of Fluency prediction.

These semantically augmented tree kernel systems can be compared to the one built with pure syntactic trees. We build such a system named SyTK/D-ST in Table 7.4 using syntactic dependency trees of the source and target side used in Section 4.3.2. Apparently, none of the augmentation methods help improve over the pure syntax-based systems in predicting human-targeted metrics. There are however slight improvements in manual metric prediction using the second method (SeTK/D-SAS$_{node}$), though none of them are statistically significant.

Overall, the overhead imposed by acquiring semantic information for the semantic-based QE systems built here does not seem to be worthwhile. We explore alternative methods of utilizing such information in the next sections.

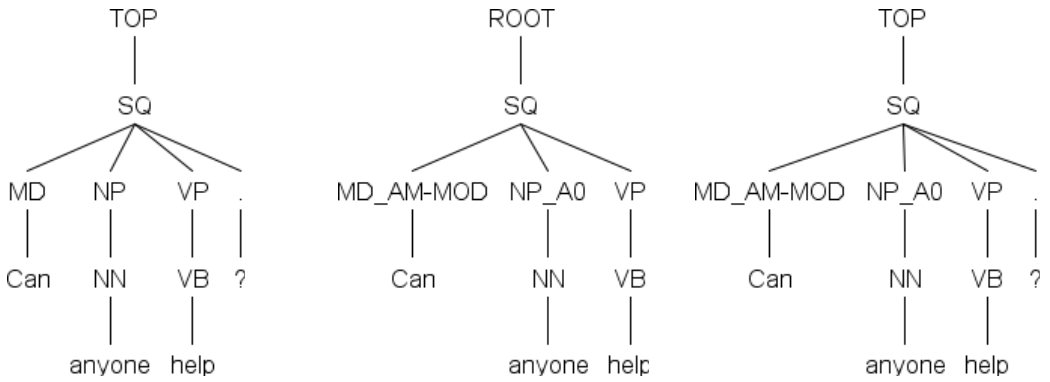
### 7.3.2 Constituency Tree Kernels

Due to the restriction of our SRL resources for the target side, we use dependency-based semantic role labelling. Semantic roles on a dependency tree are assigned to the dependent nodes of the dependency relations which are single word tokens. Focusing on the translation of a word token overlooks the rest of the phrase consisting of that word (generally as its syntactic head). Therefore, it might be helpful to convert the dependency-based semantic role labelling to constituency-based one so that the constituents can be involved in the semantic QE systems.

While constituency-based SRL can be converted to dependency formalism using head percolation rules (Surdeanu et al., 2008; Collins, 1997; Magerman, 1995), the other way around is not as straightforward. We therefore approximate the conversion using a heuristic we call (`D2C`) which transfers the semantic role labelling from the dependency tree of a sentence to its constituency tree. This heuristic recursively elevates the argument role already assigned to a terminal node in the constituency tree (based on the dependency-based argument position) to the parent node as long as the following criteria are satisfied:

1. The argument node is not a root node.
2. The role is not an `AM-NEG`, `AM-MOD` or `AM-DIS` adjunct, since this roles are normally assigned to pre-terminals.
3. The parent node does not dominate the predicate node of the argument or another argument node of the same predicate.

Once the semantic role labelling is transferred, we extract constituency-based proposition subtrees (`PST`) and semantically augmented trees (`SAS`) from them similar to the dependency tree kernels, and build the tree kernel QE systems using these new formats as described in the following sections.




(a) Constituency tree (b) Constituency PST (c) Constituency SAS

Figure 7.4: Constituency tree for the sentence *Can anyone help?* and the `PST` and `SAS` formats extracted from it. The minimal difference between (b) and (c) is specific to this example where the proposition subtree spans over the majority of the constituency tree nodes.

#### 7.3.2.1 PST Format

The constituency PSTs are the lowest common subtrees spanning the predicate node and its argument nodes and are gathered under a dummy root node for each sentence. The argument role labels are concatenated with the syntactic label of the argument node. Predicates are not marked. A sample PST is presented in Figure 7.4b which is extracted from the constituency tree of the sentence shown in 7.4a.

We build a tree kernel system using these PSTs in the same way the dependency-based PST tree kernels were built in the previous section. Table 7.5 shows the evaluation results for this system (`SeTK/C-PST`). Overall, the system performs at the same level as the dependency-based PSTs in Table 7.4. Human-targeted metric prediction scores are slightly higher while the manual metric prediction scores are lower. Therefore, as in the dependency-based experiments, we switch to the augmentation method described next.

#### 7.3.2.2 SAS Format

In the constituency SAS format, the argument role labels are affixed to the syntactic label of the argument node in the constituency tree to which the dependency-based SRL is transferred via the `D2C` heuristic. Figure 7.4c shows the semantic augmenta-

Table 7.5: Constituency tree kernel systems

| | 1-HTER | | HBLEU | | Adequacy | | Fluency | |
|---|---|---|---|---|---|---|---|---|
| | RSME | r | RSME | r | RSME | r | RSME | r |
| B-WMT17 | 0.2310 | 0.3661 | 0.2696 | 0.3806 | 0.2219 | 0.4710 | 0.2469 | 0.4769 |
| SeTK/C-PST | 0.2400 | 0.2319 | 0.2809 | 0.2541 | 0.2410 | 0.2966 | 0.2615 | 0.3616 |
| SeTK/C-SAS | 0.2289 | 0.3462 | 0.2744 | 0.3359 | 0.2261 | 0.4277 | 0.2441 | 0.4940 |
| SyTK/C-ST | 0.2292 | 0.3446 | 0.2749 | 0.3349 | 0.2266 | 0.4241 | 0.2442 | 0.4939 |

tion of the constituency tree in Figure 7.4a. The system built with the augmented constituency trees is shown in Table 7.5 as SeTK/C-SAS. As expected, it performs better than the constituency PST system. The system can also be compared with SyTK/C-ST which is the system built with the original constituency trees of the source and target; all the scores are higher but the difference are negligible.

Compared to the augmented dependency trees (SeTK/D-SAS$_{afx}$), we see a slight improvement in predicting manual metrics. However, human-targeted metric prediction, especially HTER prediction has degraded. This behaviour stems from the fact that the plain dependency-based tree kernels perform better than the plain constituency tree kernels as can be seen by comparing SyTK/C-ST in Table 7.5 with SyTK/D-ST in Table 7.4. This is curious as constituency trees contain more structure than dependency trees and have shown to be more useful in Section 4.2.2 in Chapter 4 when the News data set was used. Another possible reason is that the dependency-based argument roles transferred to the constituency trees are not accurate. Again, it is apparent that drawing too many conclusions from experiments on a single data set should be avoided.

### 7.3.3 Combined Constituency and Dependency Tree Kernels

We now combine the constituency- and dependency-based tree kernel QE systems to examine their complementarity. Two different combinations are considered: 1) combining PST-based systems, and 2) combining semantically augmented systems.

Table 7.6: Combined constituency and dependency tree kernel systems

| | 1-HTER | | HBLEU | | Adequacy | | Fluency | |
|---|---|---|---|---|---|---|---|---|
| | RSME | r | RSME | r | RSME | r | RSME | r |
| B-WMT17 | 0.2310 | 0.3661 | 0.2696 | 0.3806 | 0.2219 | 0.4710 | 0.2469 | 0.4769 |
| SyTK | 0.2267 | 0.3693 | 0.2721 | 0.3559 | 0.2258 | 0.4306 | 0.2431 | 0.5013 |
| B+SyTK | 0.2243 | 0.3935 | 0.2655 | 0.4082 | 0.2215 | 0.4632 | 0.2403 | 0.5144 |
| SeTK/CD-PST | 0.2394 | 0.2311 | 0.2795 | 0.2714 | 0.2373 | 0.3303 | 0.2578 | 0.3923 |
| SSTK | 0.2269 | 0.3682 | 0.2722 | 0.3537 | 0.2253 | 0.4351 | 0.2425 | 0.5046 |
| B+SSTK | 0.2227 | 0.4104 | 0.2671 | 0.3948 | 0.2174 | 0.4957 | 0.2381 | 0.5273 |

Table 7.6 shows the performance of the new systems.

The combination of PST systems (SeTK/CD-PST) outperforms its best component, except in predicting HTER where both systems achieve almost the same scores. The improvement in Fluency prediction is statistically significant. The performances, however, are below to the baseline.

On the other hand, the augmented combination, which is in fact our syntactico-semantic tree kernels systems thus named SSTK, improves only slightly over its highest-performing component but in all settings. The improvements for Fluency prediction are the highest amongst all albeit not statistically significant.

Compared to the plain syntactic tree kernel system also presented in the table as SyTK, manual metric prediction is marginally improved but the human-targeted metric prediction scores are slightly lower. None of these positive or negative differences are statistically significant.

Moreover, this system outperforms the baseline in only Fluency prediction. Again, the gaps are not statistically significant. When combined with the baseline, the resulting system, B+SSTK, improves over both components. The improvements in Fluency prediction scores as well as the RMSE of HTER prediction are statistically significant; other gains are not statistically significant, although they are relatively large. When compared to the combination of plain syntactic tree kernels (B+SyTK), all metrics have better predictions except the HBLEU and especially the Adequacy.

Overall, semantic tree kernels show no advantage over syntactic tree kernels

and require one more preprocessing step. However, their combination with the WMT baseline features is more fruitful than the combination of syntax-based tree kernels with these features. It should also be noted that the tree kernel approach to semantic-based QE required more engineering effort than when it was used for syntax-based QE in order to find a suitable representation. Therefore, the advantage of this method over hand-crafted features should be taken with a grain of salt. In the next section we replace tree kernels with hand-crafted semantic features.

## 7.4 Semantic-based QE with Hand-crafted Features

In Section 4.2.3 of Chapter 4, we designed a set of features extracted from the constituency and dependency parses of the source and target text, to capture various aspects of translation quality. In a similar vein, and as an alternative/complement to the semantic tree kernels of the previous section, we introduce a set of such features extracted from the semantic role labelling of the source and target text. Table 7.7 lists the feature templates in this feature set. The main idea behind this set of features is to capture the predicate-argument correspondence between the source and target as they are extracted from both source and target. The features model various aspects of this correspondence such as the role/predicate labels, role fillers as well as their syntactic annotation.

Each feature template may contain one or more features. Feature templates 1 to 5 each contain two features, one extracted from the source and the other from the translation. Feature templates 6 to 8 are nominal features and need to be binarized as we use SVM for learning the QE model. This leads to several features per feature template. Similar to the syntactic hand-crafted features in Section 4.2.3 of Chapter 4, we impose a frequency cutoff threshold to control the number of binarized features in the feature set in order to tackle the sparsity. To compute argument span sizes (feature templates 4 and 5), we use the constituency conversion of SRL

Table 7.7: Original semantic feature set to capture the predicate-argument correspondence between the source and target; each feature is extracted from both source and target, except feature number 9 which is based on the word alignment between source and target.

| | |
|---|---|
| 1 | Number of propositions |
| 2 | Number of arguments |
| 3 | Average number of arguments per proposition |
| 4 | Sum of span sizes of arguments |
| 5 | Ratio of sum of span sizes of arguments to sentence length |
| 6 | Proposition label sequences |
| 7 | Constituency label sequences of proposition elements |
| 8 | Dependency label sequences of proposition elements |
| 9 | Percentage of predicate/argument word alignment mapping types |

obtained using the `D2C` heuristic introduced in Section 7.3.2. The proposition label sequence (feature template 6) is the concatenation of argument roles and predicate labels of the proposition with their preserved order (e.g. `A0-go.01-A4`). Similarly, constituency and dependency label sequences of the proposition elements (feature templates 7 and 8) are extracted by replacing argument and predicate labels with their constituency and dependency labels respectively.

Feature template 9 consists of three features based on word alignment of source and target sentences: number of *non-aligned*, *one-to-many-aligned* and *many-to-one-aligned* predicates and arguments. The word alignments are obtained using the `grow-diag-final-and` heuristic as they performed slightly better than other types implemented in the `Moses` toolkit.

There are 62 individual features in the feature set. It should be noted that a number of features in addition to those presented here have been tried. For example, all numerical feature templates have the ratio and difference of the source and target feature values. Other examples are POS tag sequences of proposition elements similar to feature temples 7 and 8, and number of predicate/argument word alignment mapping types similar to their percentage in feature template 9. However, through a manual feature selection, we have removed features which do not appear to contribute much.

Table 7.8: QE system with hand-crafted semantic features

| | 1-HTER | | HBLEU | | Adequacy | | Fluency | |
|---|---|---|---|---|---|---|---|---|
| | RSME | r | RSME | r | RSME | r | RSME | r |
| B-WMT17 | 0.2310 | 0.3661 | 0.2696 | 0.3806 | 0.2219 | 0.4710 | 0.2469 | 0.4769 |
| SeTK/CD-PST | 0.2394 | 0.2311 | 0.2795 | 0.2714 | 0.2373 | 0.3303 | 0.2578 | 0.3923 |
| SyHC | 0.2435 | 0.2572 | 0.2797 | 0.3080 | 0.2334 | 0.3961 | 0.2479 | 0.4696 |
| SeHC | 0.2482 | 0.1794 | 0.2868 | 0.1636 | 0.2416 | 0.2972 | 0.2612 | 0.3577 |
| SSHC | 0.2362 | 0.3107 | 0.2787 | 0.3027 | 0.2326 | 0.4009 | 0.2471 | 0.4726 |
| B+SeHC | 0.2310 | 0.3660 | 0.2697 | 0.3792 | 0.2275 | 0.4444 | 0.2439 | 0.4873 |
| B+SSHC | 0.2271 | 0.4066 | 0.2677 | 0.4030 | 0.2252 | 0.4658 | 0.2393 | 0.5234 |

Table 7.8 shows the performance of the system (SeHC) built with these features. For comparison purpose, the baseline system (B-WMT17), the tree kernel system built with PST tree kernels in the previous section, (SeTK/CD-PST), as well as the systems built with syntax-based hand-crafted features (SyHC; see Section 4.3.3) are also given (in grey). The semantic features perform substantially lower than the syntactic features and thus the baseline, especially in predicting human-targeted scores. Moreover, SeTK/CD-PST outperforms this system with large gaps on human-targeted metric prediction.

Not surprisingly, relying solely on semantic-based features for estimating machine translation quality does not seem to be reasonable. As the first remedy, we mix them with syntax-based features, i.e. we combine SeHC and SyHC to build SSHC. Table 7.8 shows the scores achieved by this systems. As the results show, the combination is useful for HTER prediction, where the scores significantly improve over the syntax-based system. We also see slight improvements in the other three cases except for the Pearson r of HBLEU prediction, which has slightly degraded. However, compared to the baseline, only Fluency prediction obtains competitive scores.

These features are chosen from a comprehensive set of semantic features, so they should ideally capture adequacy better than general features. This is not the case however, probably because of the quality of the underlying semantic analysis.

We combine the semantic features (SeHC) with the baseline features. The combined system B+SeHC is shown in Table 7.8. It slightly improves over the baseline

in predicting Fluency, but performs the same in terms of human-targeted metric prediction. The gap with the baseline in Adequacy prediction is still statistically significant.

When the full hand-crafted system in SSHC is combined with the baseline, the resulting system (B+SSHC) obtains better scores than both components in all metrics except Adequacy prediction. None of the changes are statistically significant except from those of the Fluency prediction, which is surprising given the high increases in scores. It seems that the semantic features are harmful for adequacy prediction, which is precisely the notion they are supposed to capture. The fact that the fluency scores can be predicted more accurately than the adequacy scores may be related to the nature of these scores; it is easier to judge how fluent the translation is than how adequate it is in relation to the source sentence as only one factor is involved in the former. This is especially relevant to our data set since the source sentence can be ambiguous because 1) it comes from a technical domain and 2) it is user-generated.

## 7.5 Predicate-Argument Match (PAM)

As explained earlier, translation adequacy measures how much of the source meaning is preserved in the translated text and predicate-argument structure or semantic role labelling expresses a substantial part of the meaning. Therefore, the degree to which the predicate-argument structure of the source and its translation match could be an important clue to the translation adequacy, independent of the language pair used. We attempt to exploit *Predicate-Argument Match* (PAM) to create a metric that measures the translation adequacy.

Obviously, the crucial part of the PAM scoring algorithm is aligning the source and target predicates and arguments to find the matches. We try three means to accomplish this goal: 1) word alignment, 2) lexical translation table and 3) phrase translation table. The next sections describes our approaches to using each of these methods together with their evaluation.

It should be noted that there are cases in the data set where no predicate is identified by the SRL system in either source or target or both. Inspired by the observation that most source sentences with no identified proposition usually tend to be short and can be assumed to be easier to translate, and based on experiments on the dev set, we assign a score of 1 to such sentences. When no proposition is identified in the target side while there is a proposition in the source, we assign a score of 0.5.

### 7.5.1 Word Alignment-based PAM (WAPAM)

In this approach, a source predicate/argument is considered aligned to a target predicate/argument if there is a word alignment between them. Once the aligned predicates and arguments are identified, the problem is treated as one of SRL scoring, similar to the scoring scheme used in the CoNLL-2009 shared task (Hajič et al., 2009). Assuming the source side SRL as a reference, it computes unlabelled and labelled precision and recall of the target side SRL with respect to it as follows:

$$UPrec = \frac{\#\ aligned\ preds\ and\ their\ args}{\#\ target\ side\ preds\ and\ args}$$

$$URec = \frac{\#\ aligned\ preds\ and\ their\ args}{\#\ source\ side\ preds\ and\ args}$$

$$UF_1 = \frac{2 * UPrec * URec}{UPrec + URec}$$

$$LPrec = \frac{\#\ matching\ preds\ and\ their\ args\ labels}{\#\ target\ side\ preds\ and\ args}$$

$$LRec = \frac{\#\ matching\ preds\ and\ their\ args\ labels}{\#\ source\ side\ preds\ and\ args}$$

$$LF_1 = \frac{2 * LPrec * LRec}{LPrec + LRec}$$

where *preds* and *args* stand for predicates and arguments, $UPrec$ and $URec$ are unlabelled precision and recall and $LPrec$ and $LRec$ are labelled precision and recall respectively.

We obtain word alignments using the Moses toolkit, which can generate alignments in both directions and combine them using a number of heuristics. We try

Table 7.9: Performance of PAM metric scores using word alignments (WAPAM)

| | 1-HTER | | HBLEU | | Adequacy | | Fluency | |
|---|---|---|---|---|---|---|---|---|
| | RSME | r | RSME | r | RSME | r | RSME | r |
| UPrec | 0.3325 | 0.1862 | 0.3851 | 0.1721 | 0.3319 | 0.2215 | 0.4334 | 0.2363 |
| URec | 0.3249 | **0.2483** | 0.3706 | **0.2338** | 0.3213 | **0.2796** | 0.4171 | **0.2927** |
| $UF_1$ | **0.3175** | 0.2328 | **0.3607** | 0.2179 | **0.3108** | 0.2698 | 0.4033 | 0.2865 |
| LPrec | 0.4260 | 0.1571 | 0.3978 | 0.1627 | 0.3898 | 0.1926 | 0.3737 | 0.2401 |
| LRec | 0.4230 | 0.1878 | 0.3903 | 0.1928 | 0.3827 | 0.2335 | 0.3614 | 0.2759 |
| $LF_1$ | 0.4247 | 0.1784 | 0.3903 | 0.1835 | 0.3839 | 0.2225 | **0.3586** | 0.2688 |

intersection, union, source-to-target, as well as the grow-diag-final-and heuristic, but only the source-to-target results are reported here as they slightly outperform the others. Table 7.9 shows the RMSE and Pearson r for each PAM scores against not only the Adequacy scores but also the Fluency scores as well as human-targeted metric scores on the test data set.

As marked in the table, unlabelled recall achieves the best Pearson r scores across all settings, meaning that it better resembles the pattern in the translation quality scores. On the other hand, in terms of proximity of the estimations, i.e. RMSE, $F_1$ scores are the best; unlabelled $F_1$ is specifically preferable as its RMSE is the best for three of the metrics and labelled $F_1$ has the least RMSE estimating the Fluency scores. Overall, precision is the weakest measure for this purpose and unlabelled scores seem to be superior to labelled ones. The latter can be attributed to the fact that the unlabelled accuracy of the source and target (English and French) SRL are very close, establishing a balanced quality of predicate-argument structure between the source and target, which is essential to the performance of the PAM.

The Pearson r scores are higher than those of hand-crafted semantic features (SeHC in Table 7.8) for the human-targeted metrics but lower for the manual ones especially the Fluency. However, the RMSE scores are considerably larger, making the metric unsuitable for directly estimating the translation quality. We investigate the reasons behind this result in Section 7.5.4.

1: **for** each predicate in the source side **do**
2: Find the first predicate in the target side having the same label
3: **if** a match is found **then**
4: Add the predicates to *aligned predicates list*
5: **else**
6: Find a translation of the predicate token in the lexical translation table which matches any target predicate token
7: **if** a match is found **then**
8: Add the predicates to *aligned predicates list*
9: **for** each predicate pair in the *aligned predicates list* **do**
10: **for** each argument of the current source predicate **do**
11: Find the first target argument having the same label
12: Find a translation of the source argument token in the lexical translation table which matches the target argument token
13: **if** a match is found **then**
14: Add the arguments to the *aligned arguments list* of the current predicate pair
15: Calculate PAM scores using the *aligned predicates list and arguments lists*

Figure 7.5: LTPAM scoring algorithm

### 7.5.2 Lexical Translation Table-based PAM (LTPAM)

In essence, this approach considers a source predicate/argument aligned to a target predicate/argument if 1) they have the same label and 2) there is an entry in a *lexical translation table* for the source side predicate/argument token, a translation of which matches that target predicate/argument token. The exact algorithm is sketched in Figure 7.5

As can be noted in steps 6 to 8 in the algorithm, it does not solely rely on the predicate labels to align the predicates; it additionally checks the translation table to see if the target predicate is the translation of the source one.[5] Note also that since the alignment process relies on the argument role label, unlabelled scores are not meaningful with the `LTPAM` method.

The lexical translation table for this purpose is built using a parallel corpus consisting of the English-French Europarl corpus ( 2M sentence pairs), Symantec

[5]Although the predicate labels in the training data of the French SRL are based on English PropBank framesets, the French SRL system sometimes creates the predicate label based on its lemma. Consequently, the match between the source and target predicate labels is lost for such cases. Therefore, in addition to checking for label match, translation match is also checked.

Table 7.10: Performance of PAM metric scores using lexical translation table (LTPAM)

| | 1-HTER | | HBLEU | | Adequacy | | Fluency | |
|---|---|---|---|---|---|---|---|---|
| | RSME | r | RSME | r | RSME | r | RSME | r |
| LPrec | 0.3729 | 0.1674 | 0.3983 | 0.1534 | 0.3700 | 0.1523 | 0.4374 | 0.1509 |
| LRec | 0.3646 | **0.2245** | 0.3822 | **0.2110** | 0.3540 | **0.2281** | 0.4154 | **0.2208** |
| $LF_1$ | **0.3607** | 0.2089 | **0.3758** | 0.1942 | **0.3498** | 0.2045 | **0.4066** | 0.2012 |

translation memories introduced in Chapter 2 ( 860K sentence pairs), Symantec forum data used for SMT evaluation in that chapter plus some additional parallel sentences from Symantec forum data ( 3K sentence pairs).[6] The corpus contains approximately 2,860K sentence pairs and the resulting lexical translation table includes more than 4.5M entries.

Table 7.10 shows the evaluation results of the estimated scores using this approach. Consistently, recall scores are the best simulators of the quality score patterns and $F_1$ scores offer the closest estimates; precision scores are the weakest estimations. This is consistent with the WAPAM evaluation results.

Compared to the WAPAM, all the LTPAM estimations achieve higher RMSE and lower Pearson r, showing that word alignments are a better means to align source and target predicate argument structures than the lexical translation table. The gaps are specifically bigger for the case of manual metrics. This may imply that the quality of the lexical translation is lower compared to the word alignments. In the next section, we verify the extent to which this hypothesis is true.

#### 7.5.2.1 Filtering the Lexical Translation Table

In order to reduce the impact of noise existing in the lexical translation table, we filter the table using the translation probabilities assigned to each entry. We try various threshold values, below which the entries are filtered out. These values include 0.9, 0.75, 0.5, 0.25, 0.1, 0.05, 0.01, 0.005 and 0.001. It appears that high thresholds such as 0.9 and 0.75 are too restrictive. Surprisingly, however, lower

[6]We use the output of training step 4 of Moses toolkit to build the lexical translation table.

Table 7.11: Performance of PAM metric scores using filtered lexical translation table (LTPAM)

| | 1-HTER | | HBLEU | | Adequacy | | Fluency | |
|---|---|---|---|---|---|---|---|---|
| | RSME | r | RSME | r | RSME | r | RSME | r |
| LPrec | 0.3886 | 0.1837 | 0.3951 | 0.1699 | 0.3756 | 0.1739 | 0.4156 | 0.1841 |
| LRec | 0.3830 | **0.2323** | 0.3814 | **0.2221** | 0.3633 | **0.2384** | 0.3963 | **0.2445** |
| $LF_1$ | **0.3804** | 0.2190 | **0.3766** | 0.2061 | **0.3603** | 0.2183 | **0.3885** | 0.2281 |

values are also not useful; 0.5, 0.25, 0.1 do not lead to a lexical translation table which better suits our application. We resort to values below 0.1 and find 0.005 to be better than other thresholds. The results obtained by LTPAM after filtering the lexical translation table with this threshold are presented in Table 7.11

A consistent behaviour is seen with those we observed with the original lexical translation table. Compared to those results, Pearson r improves at the cost of RMSE, except for the RMSE of Fluency estimation which has improved. Nonetheless, the new scores are still lower than the WAPAM scores seen in Table 7.9.

### 7.5.3 Phrase Translation Table-based PAM (PTPAM)

Using dependency-based semantic role labelling and the lexical translation table for calculating PAM scores, only the translation of single argument-bearing tokens are taken into account, overlooking the rest of the phrase headed by that argument token. PTPAM is another PAM scoring method which tries to address this problem. This approach is similar to LTPAM with the difference being that the arguments boundaries are syntactic phrases instead of words. Therefore, *phrase translation table* is used instead of the lexical translation table. The semantic roles are transferred from word tokens to syntactic constituents for this purpose using the D2C heuristic in the same way as in Section 7.3.2.

To build the phrase translation table, we use the same parallel corpus used to build the lexical translation table in the previous section.[7] The table constructed

[7]We use the output of step 6 of the same training process used for creating the lexical translation table to build the phrase translation table.

Table 7.12: Performance of PAM metric scores using phrase translation table (PTPAM)

| | 1-HTER | | HBLEU | | Adequacy | | Fluency | |
|---|---|---|---|---|---|---|---|---|
| | RSME | r | RSME | r | RSME | r | RSME | r |
| LPrec | 0.4166 | 0.1384 | 0.4013 | 0.1253 | 0.3886 | 0.1513 | 0.3928 | 0.1707 |
| LRec | **0.4122** | **0.1910** | 0.3896 | **0.1794** | **0.3782** | **0.2184** | 0.3747 | **0.2326** |
| $LF_1$ | 0.4131 | 0.1736 | **0.3889** | 0.1602 | 0.3792 | 0.1952 | **0.3711** | 0.2141 |

in this way contains approximately 90M entries. This demands a huge memory and lookups in the table can be time-consuming, while only a small fraction of it is relevant to our data set. Therefore, we filter out those entries the source side of which are not seen in the source side of our QE data set.[8]

Table 7.12 shows the results of the evaluation of quality scores estimated using PTPAM. With this approach, recall and $F_1$ offer closer estimations than before; while the latter is still preferable in terms of Pearson r, the RMSE of their estimated scores are very close. Compared to LTPAM in Table 7.10 only Fluency estimation scores are slightly better. This degradation can be attributed to the quality of our D2C heuristic and/or the quality of phrase translation table. Considering that the heuristic is conservative and does not take too much risk in moving role labels in the constituency tree, the latter seem to be the main culprit; after all, it is expected that the phrase translation table contains more noise than the lexical translation table upon which it is built.

Similar to LTPAM, we attempt to reduce the noise from the phrase translation table relying on the translation probabilities. Based on our observation in filtering the lexical translation table, we start with average values for the probability threshold instead of high values. We specifically examine 0.5, 0.25, 0.1, 0.05, 0.01 and 0.005 and find that the estimation quality tends to increase by decreasing the threshold. The best results are obtained with 0.01 which are shown in Table 7.13

The score pattern we see here is consistent with that of the original PTPAM in

[8]We use `filter-model-given-input.pl` script in Moses toolkit.

Table 7.13: Performance of PAM metric scores using filtered phrase translation table (PTPAM)

| | 1-HTER | | HBLEU | | Adequacy | | Fluency | |
|---|---|---|---|---|---|---|---|---|
| | RSME | r | RSME | r | RSME | r | RSME | r |
| LPrec | 0.4543 | 0.1465 | 0.4217 | 0.1336 | 0.4186 | 0.1636 | 0.3924 | 0.1889 |
| LRec | **0.4511** | **0.1886** | 0.4125 | **0.1776** | **0.4106** | **0.2185** | 0.3777 | **0.2383** |
| $LF_1$ | 0.4523 | 0.1745 | **0.4120** | 0.1620 | 0.4116 | 0.1999 | **0.3742** | 0.2247 |

Table 7.12. In terms of the performance, the scores have mostly dropped, showing that the filtered phrase translation table is not useful. This suggests that phrase translation probabilities are not reliable measures of translation quality.

To conclude the experiments with PAM scoring, the word alignments seem to be the best means to align the predicates and arguments of the source and target sides of the translations. However, the resulting PAM scores are not acceptable estimations of the translation quality measured by any MT evaluation metric we use here. In the Section 7.5.5 we look at another way of utilizing the PAM scores. Before that, however, we analyse the PAM scoring in the next section to find the reasons hindering its performance.

### 7.5.4 Analysing PAM

Theoretically, PAM scores should generally be able to capture the adequacy of translation with a reasonable accuracy.[9] This is however not the case in practice in our problem setting as we saw in the previous section. There are two factors involved in the PAM scoring procedure, the quality of which can affect its performance:

- predicate-argument **structure** of the source and its translation
- **alignment** of the predicate-argument structures of the source and target

The SRL systems for both English and French are trained on edited newswire. On the other hand, our data is neither from the same domain nor edited. The

[9]There can be exceptions to this hypothesis such as when an idiom in the source text is literally translated.

Table 7.14: Results of manuals analysis of problems hindering PAM scoring accuracy

| Problem | Count | Aggregation |
|---|---|---|
| Source predicate identification | 16 | 82 |
| Source predicate labelling | 0 | |
| Source argument identification | 63 | |
| Source argument labelling | 3 | |
| Target predicate identification | 13 | 138 |
| Target predicate labelling | 9 | |
| Target argument identification | 89 | |
| Target argument labelling | 27 | |
| Alignment | 8 | 8 |
| Translation divergences | 9 | 9 |

problem is exacerbated on the translation target side, where our French SRL system is trained on only a small data set and applied to machine translation output. To discover the contribution of each of these factors in the accuracy of the PAM, we carry out a manual analysis. We randomly select 10% of the development set (50 sentences) and count the number of various problems falling in each of these two categories. The results are presented in Table 7.14

We find only 8 cases in which a wrong word alignment misleads the PAM scoring. On the other hand, there are 219 cases of SRL problems, including predicate and argument identification and labelling: 82 cases (37%) in the source and 138 cases (63%) in the target. As expected, there are more target side SRL problems than the source side. However, 82 errors in the source side within 50 sentences indicates a significant room for improvement on our English SRL system.

It can be seen that identification problems constitute more than 82% of the SRL problems. Interestingly, there are fewer target side predicate identification problems affecting PAM scores than source side. This can be related to the relatively high performance of predicate identification of French SRL shown in Table 7.2, which is very close to that of the English SRL. In the opposite perspective, this suggests that the accuracy of the PAM scores is correlated with the quality of semantic role labelling.

We additionally look for the cases where a translation divergence causes predicate-argument mismatch in the source and translation. For example, *without sacrificing* is translated into *sans impact sur* (*without impact on*), a case of *transposition*, where the source side verb predicate is left unaligned thus affecting the PAM score. We find only 9 such cases in the sample, which is similar to the proportion of the word alignment problems. This suggests that the predicate-argument structure match can be applied for estimating the quality of machine translation without worrying about the translation shifts.

As mentioned in the previous section, PAM scoring has to assign default values for cases in which there is no predicate in the source or target. This can be another source of estimation error. In order to verify its effect, we find such cases in the development set and manually categorize them based on the reason causing the sentence to be left without predicates. There are 79 (16%) source and 96 (19%) target sentences for which the SRL systems do not identify any predicate, out of which 64 cases have both sides without any predicate. When these 96 cases are taken out from the development set, the RMSE and Pearson r improve by 7% and 22% respectively.

We find that these source sentences have no predicates identified due to the following reasons:

- 57 (72%) because of sentence structure (e.g. copula verbs which are not labelled as predicates in the SRL training data, titles, etc.),
- 20 (25%) because of a predicate identification error of the SRL system
- 2 (3%) because of spelling errors misleading the SRL system

On the other hand, the reasons causing these target side sentences to have no predicate identified are as follows:

- 65 (68%) due to sentence structure
- 14 (14.5%) due a SRL error
- 13 (13.5%) due to mistranslation

- 2 (2%) due to untranslated spelling errors
- 2 (2%) due to tokenisation errors misleading the SRL system

Only mistranslation problems which comprise 13.5% of the problems can actually help PAM to capture the quality of translation. The main reason leading to the sentences without verbal predicates is the sentence structure. This problem can be alleviated by employing nominal predicates on both sides. While this is possible for the English side, there are currently no French resources where nominal predicates have been annotated.

### 7.5.5 PAM Scores as Hand-crafted Features

An alternative way to employ the PAM scores in estimating the machine translation quality is to use them as features in a statistical framework. Due to their higher performance when evaluated directly, `WAPAM` scores are used.[10] Similar to our hand-crafted QE systems in this work, we build a SVM model using all 6 `WAPAM` scores. The performance of this system ($\texttt{SeHC}_{pam}$) on the test set is shown in Table 7.15. The performance is considerably higher than when the PAM scores are used directly as estimations. For comparison purpose, the hand-crafted semantic system `SeHC` from Section 7.4 is also given in the table. Interestingly, compared to the 62 semantic hand-crafted features of `SeHC`, this small feature set performs noticeably better in predicting human-targeted metrics. However, despite the big gaps, only RMSE difference of HTER prediction is statistically significant. On the other hand, for the manual metrics, although the performance of this feature set is lower than the `SeHC`, only the RMSE difference of Fluency prediction is statistically significant.

We add the new features to our set of hand-crafted features in `SeHC` to yield a new system named $\texttt{SeHC}_{+pam}$ in Table 7.15. As can be seen, all scores improve compared to the stronger of the two components, except for RMSE of HTER prediction. The gain from the combination is particularly considerable and statistically significant

[10] We also tried `LTPAM` and `PTPAM` scores; their performances vary in the same way they did when used directly as estimations.

Table 7.15: Performance of WAPAM scores as features, alone ($\texttt{SeHC}_{pam}$) and combined ($\texttt{SeHC}_{+pam}$))

| | 1-HTER | | HBLEU | | Adequacy | | Fluency | |
|---|---|---|---|---|---|---|---|---|
| | RSME | r | RSME | r | RSME | r | RSME | r |
| B-WMT17 | 0.2310 | 0.3661 | 0.2696 | 0.3806 | 0.2219 | 0.4710 | 0.2469 | 0.4769 |
| SeHC | 0.2482 | 0.1794 | 0.2868 | 0.1636 | 0.2416 | 0.2972 | 0.2612 | 0.3577 |
| $\texttt{SeHC}_{pam}$ | 0.2414 | 0.2292 | 0.2833 | 0.2195 | 0.2414 | 0.2787 | 0.2661 | 0.3210 |
| $\texttt{SeHC}_{+pam}$ | 0.2445 | 0.2387 | 0.2822 | 0.2368 | 0.2370 | 0.3571 | 0.2575 | 0.3908 |
| $\texttt{B+SeHC}_{pam}$ | 0.2274 | 0.3977 | 0.2666 | 0.4069 | 0.2198 | 0.4854 | 0.2419 | 0.5016 |
| $\texttt{B+SeHC}_{+pam}$ | 0.2337 | 0.3417 | 0.2701 | 0.3697 | 0.2224 | 0.4694 | 0.2439 | 0.4881 |

in the case of manual metric prediction. However, the performance is still not close to the baseline.

We combine both the PAM feature set ($\texttt{SeHC}_{pam}$) and the semantic feature set ($\texttt{SeHC}_{+pam}$) with the baseline features separately. The results are shown in Table 7.15 as well. Interestingly, the gains from combining the smaller feature set with the baseline ($\texttt{B+SeHC}_{pam}$) is larger. In fact, the larger combination ($\texttt{B+SeHC}_{+pam}$) degrades except in Fluency prediction. Therefore, PAM features can replace a larger and thus costlier semantic feature set in a real world application where features of various genres are combined to build a quality estimation system.

## 7.6 Combined Semantic-based QE System

In Section 4.3.4 of Chapter 4, we combined syntactic tree kernels and hand-crafted features to build SyQE, our syntax-based QE system. In this section, we build the semantic counterpart of that system by combining the semantic tree kernel system (SSTK) and the semantic hand-crafted system including PAM features ($\texttt{SeHC}_{+pam}$). This system, is named SeQE in Table 7.16. Individual systems, as well as the baseline and the syntax-based system (SyQE) are also given (in gray) in the table for comparison purposes.

SeQE performs better than the stronger of its components. Except for adequacy prediction, the other improvements are statistically significant. It also slightly out-

Table 7.16: Performance of semantic-based QE system and its combination with the baseline

| | 1-HTER | | HBLEU | | Adequacy | | Fluency | |
|---|---|---|---|---|---|---|---|---|
| | RSME | r | RSME | r | RSME | r | RSME | r |
| B-WMT17 | 0.2310 | 0.3661 | 0.2696 | 0.3806 | 0.2219 | 0.4710 | 0.2469 | 0.4769 |
| SyQE | 0.2255 | 0.3824 | 0.2711 | 0.3650 | 0.2248 | 0.4393 | 0.2419 | 0.5087 |
| SSTK | 0.2269 | 0.3682 | 0.2722 | 0.3537 | 0.2253 | 0.4351 | 0.2425 | 0.5046 |
| $\text{SeHC}_{+pam}$ | 0.2445 | 0.2387 | 0.2822 | 0.2368 | 0.2370 | 0.3571 | 0.2575 | 0.3908 |
| SeQE | 0.2249 | 0.3884 | 0.2710 | 0.3648 | 0.2242 | 0.4447 | 0.2404 | 0.5182 |
| B+SeQE | 0.2219 | 0.4194 | 0.2670 | 0.3975 | 0.2188 | 0.4882 | 0.2362 | 0.5427 |

performs SyQE for all metrics other than HBLEU. Compared to the baseline, we see mixed results: HTER and Fluency prediction scores are higher than the baseline but BLEU and Adequacy prediction scores are lower. However, among all the changes, only the Fluency prediction improvements are statistically significant.

In addition, we examine the complementarity of our semantic-based system with the baseline features by combining SeQE and B-WMT17. The new system is B+SeQE and shown in Table 7.16. It is the highest-performing system built on this data set so far in this work and outperforms both of its components; all the gaps with SeQE and HTER and Fluency prediction gaps with the baseline are statistically significant.

## 7.7 Syntactico-semantic-based QE system

Finally, we build our full syntactico-semantic quality estimation system. This system is the combination of the syntax-based and semantic based QE systems in SyQE and SeQE respectively. It should be noted that these two systems are combined without syntactic tree kernels (SyTK in Section 4.3.2) to avoid redundancy with SSTK, the tree kernel component of SeQE, as these are the augmented syntactic tree kernels. Table 7.17 shows the performance of this system named SSQE. It can be compared to the baseline and each of its components in the same table.

The full syntactic-semantic system (SSQE) improves over its syntactic and semantic components, though the improvements are not statistically significant. Compared

Table 7.17: Performance of syntactico-semantic QE system and its combination with the baseline

| | 1-HTER | | HBLEU | | Adequacy | | Fluency | |
|---|---|---|---|---|---|---|---|---|
| | RSME | r | RSME | r | RSME | r | RSME | r |
| B-WMT17 | 0.2310 | 0.3661 | 0.2696 | 0.3806 | 0.2219 | 0.4710 | 0.2469 | 0.4769 |
| SyQE | 0.2255 | 0.3824 | 0.2711 | 0.3650 | 0.2248 | 0.4393 | 0.2419 | 0.5087 |
| SeQE | 0.2249 | 0.3884 | 0.2710 | 0.3648 | 0.2242 | 0.4447 | 0.2404 | 0.5182 |
| SSQE | 0.2246 | 0.3920 | 0.2696 | 0.3768 | 0.2230 | 0.4538 | 0.2402 | 0.5196 |
| B+SSQE | **0.2225** | **0.4144** | **0.2673** | **0.3953** | **0.2202** | **0.4771** | **0.2379** | **0.5331** |

to the baseline, HTER and Fluency prediction perform better, the latter being statistically significant. HBLEU prediction is around the same as the baseline, but Adequacy prediction performance is lower, though not statistically significantly. This is not the first time we observe this pairing of the metrics: HTER with Fluency and HBLEU with Adequacy.

The last QE system we build is the combination of our syntactic-semantic system with the baseline features. This combination improves over its components; compared to the stronger component, only the HTER and Fluency prediction improvements are statistically significant. However, compared to the combination of semantic-based system with the baseline features in B+SeQE, the results are slightly lower. Recall that this system contains only the syntactic hand-crafted features (SyHC) further compared to B+SeQE. Therefore, these features have a diminishing effect on B+SSQE. Interestingly, the small Fluency and Adequacy prediction scores gaps between these two systems are statistically significant. Regardless of this comparison, the results confirm the synergy between syntactic/semantic information and general surface-driven features we have been observing throughout our QE experiments.

## 7.8 Summary and Conclusion

We investigated various approaches to using semantic information in quality estimation of machine translation. The particular semantic representation we tried here

was semantic role labelling or predicate-argument structure. Using tree kernels, we found that purely semantic tree kernels built using proposition subtrees did not lead to reliable systems comparable to the baseline. Alternatively, augmenting syntactic tree kernels built in Chapter 4 with semantic role labelling showed to be a better approach. Although significant benefit was not achieved over the plain syntactic tree kernels, their combination with the baseline features was found to be more fruitful than the combination of the syntactic tree kernels with these features.

We then designed a comprehensive set of hand-crafted features extracted from the semantic role labelling of the source and target text. This feature set did not capture enough information about the quality of translation; it performed considerably lower than the semantic tree kernels.

In addition to these statistical methods, we defined a metric, PAM, for estimating the translation quality based on predicate-argument structure match between source and target. Unlike similar metrics, PAM is computed by a simple formula and has no parameters to be tuned. We offered three variations of this metric each using different means to align the predicates and arguments of source with translation: 1) word alignments, 2) lexical translation table and 3) phrase translation table. We found word alignments more suitable for this purpose. This metric showed competitive performance with semantic hand-crafted features in terms of correlation with the quality scores. However, the estimation errors were much higher. We found that word alignment and translation divergence only have minor effects on the performance of this metric, whereas the quality of semantic role labelling is the main hindering factor. Another major issue affecting the performance of PAM is the unavailability of nominal predicate annotation.

Using the PAM scores as features in a statistical framework led to better estimations than directly using them as quality scores. These features alone outperformed the whole set of semantic hand-crafted features in predicting human-target metrics. When combined, these two sets of features yielded a better system.

Our syntactico-semantic QE system could outperform the baseline in some set-

tings. However, the best QE performance accomplished in this thesis was when the hand-crafted syntactic features were removed from this system, i.e. when the baseline features were combined with the semantic hand-crafted and semantic-augmented tree kernels.

The semantic role labelling we used was based on syntactic dependency trees, in which the labels are assigned to word tokens. Since there are no constituent SRL resources available for French, we resorted to a simplistic heuristic to transfer the labels upwards in the constituency trees to phrases. Perhaps genuine constituent-based arguments can make a more accurate comparison between the source and target predicate-argument structure possible.

As mentioned, the suboptimal quality of our semantic role labelling system due to various reasons, including: 1) lack of resources for French SRL, 2) using out-of-domain parsers and semantic role labellers and 3) applying them on unedited and machine translated text, is the main culprit in the low performance of our semantic-based QE approaches. In the next chapter, we further investigate the second and third reasons by attempting to improve the parsing accuracy and measuring its effect on the semantic-based QE approaches.

# Chapter 8

# The Foreebank: Forum Treebank

In the previous chapter, we observed that the low quality of semantic role labelling was the main factor impeding the performance of the PAM quality estimation metric. The semantic role labelling systems we used were trained on resources built on edited financial newswire text. However, we applied them to unedited user-generated text from a security software support context and its machine translation. These variations, i.e. edited vs. unedited and financial newswire vs. security software support, are generally known as *domain shifts* in natural language processing and machine learning, and impose difficulties in many tasks in these fields (Daumé III and Marcu, 2006; Blitzer et al., 2006; McClosky, 2010; Banerjee, 2013). These problems mainly originate in the lack of sufficient hand-crafted resources for training machine learning models for every single domain. Most of these resources are usually created for one domain only, as they are laborious and costly to create. For example, for many years, the WSJ portion of the Penn Treebank was the only major treebank used for English parsing.

It should however be noted that the notion of domain is not well-defined in this context. It has been used at a very broad level to distinguish between written and spoken language, i.e. register (Finkel and Manning, 2009), or at a more specific level to discern between different genres of text in the Brown corpus such as press/reportage, press/editorial, religion, etc., or between different web text cat-

egories such as emails, newsgroups or user reviews (Petrov and McDonald, 2012; Mott et al., 2012). Moreover, depending on the type of variation, different terms have been used in the literature to address the distinction between the variants, among which are text genre, text type (Gildea, 2001), topic and style (Alumäe and Kurimo, 2010).

The difficulties caused by domain shift are due to various new phenomena appearing in the application context which were not seen in the contexts upon which the systems were built. When the shift occurs from edited to unedited text, these phenomena include writing errors such as spelling, capitalization, grammar, punctuation, and writing style such as informal structures, innovative use of language and emoticons. On the other hand, when moving from the financial newswire to security software support, they include vocabulary and syntactic constructions such as questions, imperatives and sentence fragments.

It has been previously shown that the performance of semantic role labelling is dependent on its underlying syntactic analysis (Punyakanok et al., 2008). The semantic role labelling system we used in the previous chapter is largely built upon syntactic features such as those derived from POS tagging of the predicate and argument and their neighbouring tokens, and from the dependency parsing of the sentence (Björkelund et al., 2009). Therefore, an improvement in the quality of syntactic parsing of the quality estimation data is likely to be translated into an improvement in the quality of its semantic role labelling. This improvement can in turn enhance the performance of semantic-based quality estimation systems, especially the PAM metric which uses this information directly. For example, a simple mistake in the POS tag of the verbs in the example in Figure 8.1 leads to the SRL system losing/mislabelling predicates and the miscalculation of the PAM scores in turn. While the adequacy score assigned by the human evaluator is 0.75 (when scaled on the [0-1] range), the URec (unlabelled recall) and $UF_1$ (unlabelled $F_1$) WAPAM scores (see Section 7.5.1 of Chapter 7) are 0.2857 and 0.3077.[1] Fixing these POS

[1] URec and $UF_1$ are used here because they were the best performing scores according to the

| | ID | Alignment | Word form | POS | Predicate | profile.01 | restart.01 |
|---|---|---|---|---|---|---|---|
| Source | 1 | 1 | ( | -LRB- | _ | _ | _ |
| | 2 | 5 | -WFP- | PRP | _ | A0 | A0 |
| | 3 | 3 | profile | VB | profile.01 | _ | _ |
| | 4 | 7 | -WFP- | PRP | _ | A1 | _ |
| | 5 | 8 | ) | -RRB- | _ | _ | _ |
| | 6 | 9 | Delete | IN | _ | AM-LOC | _ |
| | 7 | 10 | the | DT | _ | _ | _ |
| | 8 | 11 | three | CD | _ | _ | _ |
| | 9 | 12 | files | NNS | _ | _ | _ |
| | 10 | 13 | and | CC | _ | _ | _ |
| | 11 | 14 | then | RB | _ | _ | _ |
| | 12 | 15 | restart | VBP | restart.01 | _ | _ |
| | 13 | 16 | FireFox | NNP | _ | _ | A1 |
| | 14 | 17 | . | . | _ | _ | _ |

| | ID | Alignment | Word form | POS | Predicate | remove.01 | redémarrez.01 |
|---|---|---|---|---|---|---|---|
| Target | 1 | 1 | ( | PONCT | _ | _ | _ |
| | 2 | | - | PONCT | _ | _ | _ |
| | 3 | 3 | Profil | NPP | _ | A0 | _ |
| | 4 | | de | P | _ | _ | _ |
| | 5 | 2 | WFP | NPP | _ | _ | _ |
| | 6 | | - | PONCT | _ | _ | _ |
| | 7 | 4 | WFP | NPP | _ | _ | _ |
| | 8 | 5 | ) | PONCT | _ | _ | _ |
| | 9 | 6 | supprimez | V | remove.01 | _ | _ |
| | 10 | 7 | les | DET | _ | _ | _ |
| | 11 | 8 | trois | ADJ | _ | _ | _ |
| | 12 | 9 | fichiers | NC | _ | A1 | _ |
| | 13 | 10 | et | CC | _ | _ | _ |
| | 14 | 11 | puis | ADV | _ | _ | AM-ADV |
| | 15 | 12 | redémarrez | V | redémarrez.01 | _ | _ |
| | 16 | 13 | FireFox | NPP | _ | _ | A1 |
| | 17 | 14 | . | PONCT | _ | _ | _ |

Figure 8.1: Semantic role labelling of the sentence *(-WFP- profile -WFP) Delete the three files and then restart Firefox.* and its machine translation by the SRL systems in CoNLL-2009 format: 1) *profile* is mistakenly labelled as the predicate due to a wrong POS tag, 2) *Delete* is missed by the SRL system due to a wrong POS tag losing the match with *supprimez* despite a correct alignment.

tags manually and redoing the SRL changes these scores to 0.8 and 0.7273, very close to the human evaluation score.

Toward this end, we build *Foreebank*, treebanks taken from the English and French Norton forum text, from which the SymForum quality estimation data is also selected.[2] These treebanks enable us to evaluate the quality of syntactic parsing of the forum text. In addition, we adapt an annotation strategy which makes it possible to analyse the user errors in the forum text from different perspectives such as their

results in Table 7.9 in Chapter 7.

[2]There is no overlap between the Foreebank and SymForum sentences.

effect on parsing performance. This intrinsic evaluation will also help measure the impact of parsing improvement on semantic-based QE on this domain.

Foreebank comprises two data sets, one in English and one in French. While the entire English Foreebank is selected from the Norton forum, only half of the French Foreebank comes from the French Norton forum and the other half are human translations of the English forum segments. By choosing the French Foreebank from a mixture of forum text and its human translation, we aim at being as close as possible to machine translation output of French text, which is the use case in this thesis, while avoiding the difficulties of annotating machine translation output. In addition, the treebank built in this way can also be used for more general purposes in parsing French use-generated content not specific to the work in this thesis.

In the rest of this chapter, we first review the existing syntactic treebanks as well as the literature on parser domain adaptation in Section 8.1. In Section 8.2, we describe the annotation process of the Foreebank. In Section 8.3, we analyse the Foreebank by extracting some statistics from the data set. In Section 8.4, the parsing performance on the forum text is evaluated using the Foreebank. In Section 8.5, the effect of user errors in parsing the forum text is verified. In Section 8.6, we conduct a set of experiments dedicated to improving the performance of parsing forum text. In Section 8.7, we carry out the extrinsic parser evaluation by using the new parses to obtain the semantic role labelling of the quality estimation data which are then used to replicate some of the semantic-based quality estimation settings of Chapter 7.[3]

## 8.1 Related Work

Syntactic annotation of a corpus with the aim of building ignore training (and evaluation) resource for statistical parsers is a tedious task and requires not only human labour but also expertise. For this reason, only a few such corpora, which

[3]For easy comparison, the results of the replicated semantic-based QE experiments are also presented in Table A.1 in Appendix A beside other results.

are known as *treebanks*, are available for only a handful of languages. For English, the major treebank used for training parsers is the *WSJ* portion of *Penn Treebank* (Marcus et al., 1994), which contains about 45K sentences (over 1 million words) from the Wall Street Journal annotated with syntactic constituents, an American news paper with a special focus on business and economics. The WSJ corpus is formed by selecting 2,499 stories from a total of about 100K stories of 1989 Wall Street Journal. In addition to the WSJ, the Penn Treebank (PTB) includes the *Brown* corpus (Kuĉera and Francis, 1967), the *Switchboard* corpus of telephone conversations (Taylor, 1996) and a sample of the *ATIS* (Air Travel Information System) corpus (Hemphill et al., 1990) annotated with the same syntactic structure. The Brown corpus originally contains over 1 million words from 500 samples (slightly over 2000 words each) across 15 text categories (genres) selected from a variety of contemporary American English sources. These genres include press reportage, editorial and reviews, religion, skills and hobbies, lore, biographies and memories, US government, science, general, mystery, adventure, romance and science fiction and novels.

The other corpora annotated based on the Penn Treebank annotation strategy are the *Penn BioMedical Treebank* (Warner et al., 2004), the *English Translation Treebank* (Mott et al., 2009) and the *English Web Treebank* (Mott et al., 2012). The English Translation Treebank (ETTB) consists of two treebanks, the English-Chinese Treebank (ECTB) and the English-Arabic Treebank (EATB), which annotate the Chinese and Arabic newswire sentences and their translations to English in parallel. The English Web Treebank (aka Google Web Treebank) is a corpus of over 250K words, selected from weblogs, newsgroups, emails, local business reviews and Yahoo! answers. The annotation strategy for this treebank is based on Penn Treebank annotation guidelines as well as the Switchboard and other aforementioned treebanks built upon the Penn Treebank guidelines. However, the guidelines are adapted to address the phenomena specific to this type of text, which differs from the WSJ in various aspects including being user-generated, unedited and extracted

from Web pages (Petrov and McDonald, 2012). The Foreebank corpus is closer to the English Web Treebank in the same aspects. Other similar English treebanks to the Foreebank include the small treebank described in Foster et al. (2011a) which contains annotated sentences from tweets and a sports discussion forum. The annotation of these treebanks as well follows the PTB guidelines.

There has also been work addressing the parsing of question structures. Judge et al. (2006) create the QuestionBank, a set of 4,000 questions annotated with their phrase structure trees. The questions come from two different sources: the question-answering (QA) data set used to evaluate QA systems by the Text Retrieval conference[4] (TREC) and the CCG question classification data set (Li and Roth, 2002).

For French, the most widely used treebank for training parsers is the *French Treebank* (Abeillé et al., 2003). This treebank consists of over 12,000 sentences selected from the Le Monde newspaper. In addition, the *French Social Media Bank* developed by Seddah et al. (2012) is a tree bank of *noisy* user-generated data comprising 1,700 sentences from various type of social media including Facebook, Twitter, video games and medical discussion forum. To make the corpus farther from FTB, they search for sentences with some UGC-specific patterns and also add extra noise to some sentences. The annotation of this treebank is based on the FTB-UC[5] (Candito and Crabbé, 2009) annotation guidelines, extending them to suit the noisy user-generated content.

The problem of hand-crafted resource shortage is omnipresent in all areas of natural language processing and machine learning including parsing. Gildea (2001) finds that the performance of a parser trained on the WSJ is 5.7 $F_1$ points lower when evaluated on the Brown corpus than when tested on the WSJ. When the parser is trained on the Brown corpus itself, it obtains a higher $F_1$ by 3.5 points. He observes that training on a small amount of data which matches the test data is

[4] `http://trec.nist.gov/`

[5] FTB-UC is a modified version of the FTB.

more useful than training on a large amount of different data. He suggests that some features used to build the statistical parsers such as lexical co-occurrence contribute to the dependence of the parser performance on its training data.

McClosky et al. (2010) evaluate parsers trained on several different English treebanks including the WSJ, Brown, Switchboard, ETTB and GENIA (Tateisi et al., 2005) on each other and on the British National Corpus (Foster and van Genabith, 2008). The best performances on all test sets are achieved by the parsers trained on the same treebank, except for ETT which is best parsed by the WSJ-trained parser probably because it also comes from newswire. The largest performance drop due to the cross-domain application of the WSJ-trained parser is on the Switchboard test set. This can be attributed to the fact that these two treebanks not only differ in their domain but also in their register; one in edited written language and one in transcribed spoken language.

Foster (2010) investigates the performance of parsing unedited user forum text by parsing the sports forum treebank mentioned above with a model trained on the WSJ. The performance drops by 19 $F_1$ points compared to parsing the WSJ test set. This gap consistently exists across the evaluation of four different parsing systems trained on WSJ and applied to these two data sets. She manually examines the output of the parser to find the phenomena affecting the parser performance. Coordination is found to be one of the main issues due to problems such as omission of conjunctions or their replacement with comma, while the spelling errors have only a small effect on the performance.

Foster et al. (2011a) compare four different WSJ-trained parsers, two constituency and two dependency, on the sports forum and Twitter treebanks. They find that tweets are harder to parse despite their shorter length. The POS tagging errors are found to be an important contributing factor in the parsing performance of tweets.

Seddah et al. (2012) evaluate an FTB-trained parser on both the FTB test set and the French Social Media treebank. The parser performs about 20 $F_1$ points lower on the latter. Furthermore, they find that this parser performs considerably lower

on the French Social Media than on the French biomedical data (Emea French test set), which also contains many unknown words and constructions. They conclude that, despite what is generally believed, POS tagging and parsing are not close to being solved problems.

To tackle these issues, a considerable amount of research has been devoted to *domain adaptation*, the goal of which is to train models which generalize well to a new domain (Blitzer et al., 2006; Foster et al., 2008; Daumé et al., 2010; Plank, 2011). Again, the term domain is used in its broadest sense here. The simplest approaches involve adding whatever additional training data is available in the *target domain*, the domain to which the model will ultimately be applied, to the original training data. For instance, Gildea (2001) trains a parsing model by adding to the WSJ training data the Brown corpus, which is half the size of the WSJ corpus. Parsing the Brown test set with this model improves over the performance obtained by the model trained only on WSJ by 3.7 $F_1$ points. On the other hand, the combination is not very useful when its performance is compared to that of either the Brown-trained model on the Brown test set or the WSJ-trained model on the WSJ test set, indicating that adding supplementary data from a different domain than the target is not helpful regardless of its size. It should however be noted that the amount of data available for the target domain in real life may not be as large as the Brown corpus used here (over 20K sentences). Therefore, it is interesting to know how much data from target domain can contribute to what extent to the parsing accuracy. We investigate this question as part of our experiments with Foreebank in this chapter.

Petrov and McDonald (2012) describe a shared task organized to evaluate the robustness of parsing systems to the domain changes and to the noise introduced by web text. The evaluation data was the English (Google) Web Treebank described earlier. The best-performing systems used system combination methods and achieved $F_1$ scores between 80 to 84, where the baseline was in the 75 to 80 range. They find the POS tagging of Web text particularly challenging performing just above 90% of accuracy, and argue that improving POS tagging can improve

parsing, especially dependency parsing. They also find that better WSJ parsing leads to better Web parsing. They suggest that these behaviour may be an artefact of system combination and that domain adaptation is ultimately required. In other words, further improvement of in-domain parsing may not be the optimal solution to the improvement of out-of-domain parsing.

*Self-training* (Yarowsky, 1995) is one of the most well studied approaches to parser domain adaptation. McClosky et al. (2006) parse a large *unlabelled* corpus of news article from the North American News Text corpus (NANC) using the two-stage reranking parsing system of Charniak and Johnson (2005) trained on the WSJ. They then select subsets of varying sizes from these parses and mix them with the WSJ training data which is then used to retrain (i.e. self-train) the first-stage parser of the same parsing system. The self-trained parsing model in this way leads to better parses than the original model. Foster et al. (2011b) compare the use of edited sports news article with user-generated sport forum sentence as unlabelled data in the parser self-training process. They find, contrary to their expectations, that the unlabelled forum data is more useful than the unlabelled edited data for self-training. Self-training was also used by the best-performing system submitted to the Web parsing shared task (Le Roux et al., 2012), where the unlabelled web text from the same domain as the evaluation sets is used for self-training. They use a parser accuracy predictor (Ravi et al., 2008) to filter the parsed unlabelled data added to the training set of the self-trained parser. Their method trains different parsing models for each of the five genres in the Web Treebank and use each for parsing the test sentences from the corresponding genre. The genre of the test data is predicted using a classifier trained for this purpose.

To address the problem of parsing ungrammatical text, Foster et al. (2008) introduce artificial errors into WSJ sentences and then train on the parse trees of this ungrammatical sentences. The errors include agreement errors, real word spelling errors, verb form errors as well as word deletion and insertion errors. One or two errors are introduced into each sentence and the gold-standard parse tree of the sen-

tence is minimally edited to match the new sentence keeping in mind that "a truly robust parser should return an analysis for a mildly ungrammatical sentence that remains as similar as possible to the analysis it returns for the original grammatical sentence". The resulting parsing model trained on this data is able to improve over a model trained on the original WSJ in parsing the erroneous WSJ test set created in the same way by up to 3.7 $F_1$ points. Foster (2010) applies a similar approach in parsing the sports forum data set and finds significant improvement.

Seddah et al. (2012) extend the FTB tag set to annotate the contractions and typographic diaeresis phenomena (e.g. using *c a dire* for *c'est-à-dire*) of the social media. The extensions are largely consistent with the English Web Treebank addendum to the Penn Treebank annotation guidelines. They treat the corpus as two parts: less and highly noisy. For the highly noisy part, they first try to automatically reduce the noise by, for example, merging split emoticons, URLS, etc. The corrected sentences are then POS-tagged. Finally, the corrected tokens are mapped to the original ones and the POS tags are transferred to them.[6] This method significantly improves the POS tagger output.

The annotation of Foreebank is based on the PTB annotation for English and the FTB annotation for French. It also borrows from the English Web Treebank annotation guidelines to address some of the similar issues. However, it uses a novel method to annotate user errors on the parse trees. Our preliminary experiments on improving the parser performance of the Foreebank text here is based on using supplementary training data from target domain which is similar to the method used by Gildea (2001).

[6]When one-to-many maps are encountered, all the original tokens are tagged with `Y`, a special token they introduce except the last one which is assigned the real POS tag. In many-to-one cases, the POS tags are merged with + and assigned to the original token.

## 8.2 Building the Foreebank

In order to build the English Foreebank, we randomly select 1000 segments from a large collection of English Norton forum text containing 3 million segments.[7] For the French Foreebank, 500 segments are randomly selected from the 40K French Norton forum segments and 500 from a set of 3000 human-translated English Norton forum segments to French. The segmentation of the original forum text (excluding the 500 translated French segments) and the tokenisation of the resulting segments are done in the same way as the SymForum data (explained in Section 3.2.1 of Chapter 3) for both data sets.

To prepare the data for annotation, the English Foreebank segments are parsed using the Stanford parser (Klein and Manning, 2003) with its built-in lexicalized PCFG parser. We choose this parser because it uses a different algorithm compared to the Lorg parser, and thus their output is more likely to be different, which helps avoid bias in the evaluation of the parsing models used in this work. The French Foreebank segments are however parsed using the Berkeley parser as the Stanford parser did not perform well when evaluated on the FTB evaluation sets.[8]

Once the data is parsed, it is given to human annotators to correct the parses.[9] The annotation guidelines for the English Foreebank are built on the *Penn Treebank* annotation guidelines (Bies et al., 1995) and those for French on the French Treebank guidelines (Abeillé et al., 2003). To handle the phenomena specific to this text type, we either adopt the English Web Treebank guidelines (Mott et al., 2012) or develop our own strategy. These specifications are described in the following sections. We first describe the preprocessing steps and addressing problems specific to this type of text. We then explain our approach to annotating erroneous structures introduced by users.

[7] These segments come from the same source as the SymForum data (see Section 3.2.1 of Chapter 3) but do not overlap with them.

[8] The best performance achieved using various grammars and options of the Stanford parser was 73.59 $F_1$ points compared to 83.01 points for the Berkeley parser.

[9] Each of English and French data sets are annotated by a computational linguist who is the native speaker of the corresponding language.

### 8.2.1 Handling Preprocessing Problems

Natural language ambiguity makes a perfect sentence segmentation and tokenisation almost impossible when performed automatically. User errors, in addition, can both exacerbate the ambiguity and add their own problems. Furthermore, problems are imposed by text styling, especially when it is extracted from the Web. To account for real world scenarios, where such problems are inevitable and cannot be automatically fixed, our approach throughout the annotation process is to avoid manually correcting the preprocessing errors caused by user errors or text styling. Instead, we consider these issues as characteristics of user-generated content and handle them in the parse trees where possible. This is one aspect where Foreebank annotation deviates from English Web Treebank annotation which chooses to manually correct these errors before annotation.[10]

Sentence segmentation is one of the preprocessing steps which can impose challenges when run on this type of text. Since sentence boundary is vital in parsing, these challenges can affect parsing performance. User-generated errors (e.g. punctuation errors, fusions, etc.), ambiguous punctuation (e.g. full stops and abbreviation periods), text styling (e.g. an address on multiple lines) and the style of text extracted from HTML are all examples of contributing factors to such problems. Regardless of what causing the segmentation problems, they can be categorized into *merged* sentences and *split* (broken) sentences.

Merged sentences are cases where there are multiple sentences in a segment ending – or which must end – with a final punctuation, i.e. full stop, exclamation mark or question mark. In the following example, the line should have been split at *When*, but due to the punctuation error, i.e. using comma instead of full stop after *start*, the sentence segmenter has been confused.

(1) `7. Combofix will start, When it is scanning don't move the mouse cursor inside the box, can cause freezing.`

[10]When the instructions are not sufficient or applying these methods is not possible, the segment is dismissed and replaced with another segment to retain the intended data set size.

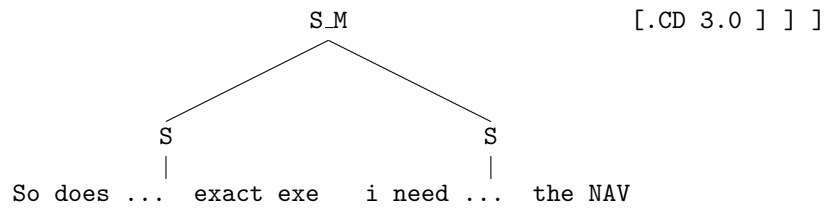



Figure 8.2: Annotation of Example 1 (The sentences are not fully shown due to space limit.)

We do not split these merged sentences unlike the English Web Treebank. Instead, the parse trees of all of them are gathered under a node with the appropriate label (e.g. `S`) suffixed with `_M`. This is illustrated in Figure 8.2, where the high-level annotation of Example 1 is displayed.

Split sentences are cases where a sentence is split over multiple lines. This can occur due to styling reasons, as in example 2 below, where the questions are listed in separate lines, or because of ambiguity, as in example 3, where the period at the end of *devs* has been interpreted as a full stop by the segmenter.

(2) `The questions to Symantec:`

(3) `I'm sure the devs.`

`can give you more details on this`

While the English Web Treebank manually joins the sentence segments spread over multiple lines, such as list items, before annotation, we annotate each split part separately if it independently conveys a comprehensible message. Otherwise, the sentence is dismissed from the treebank altogether. For instance, Example 2 can be parsed regardless of what the questions are. On the other hand, Example 3 is not a self-standing expression and is consequently not annotated.

Another important factor in parsing is the token boundaries inside a sentence. Similar to the segmentation problems, tokenisation problems can be categorised as *merged* (fused) tokens (Banerjee et al., 2012) and *split* (broken) tokens. Merged tokens are considered as a mixture of spelling errors and deleted tokens. For instance, in Example 4 whenI is considered as a spelling error for *when* and *I* as a deleted

token. The annotation of spelling errors and deleted tokens are explained in the next section.

(4) `whenI tried to use ...`

On the other hand, split tokens are grouped into two types. The first type includes morphologically broken tokens where a token is split into its morphological components as in Example 5, where *LiveUpdate* has been mistakenly broken to *Live Update*:

(5) `problem with Live Update`

In such cases, both parts are POS tagged with their correct tag and are suffixed with `_B` standing for broken. All other splits, are considered as the second type and treated as a mixture of spelling error and extraneous token which are described in the next section. For instance, *In box* in Example 6 below is the split of *Inbox* by mistake and *i t* in Example 7 is the split of *it*. *In* is considered as a real word spelling error and the *box* as an extraneous token. On the other hand *i* is treated as a spelling error and *t* as an extraneous token.

(6) `the In box had just the emails from ...`

(7) `i t keeps causing Norton to lock up ...`

### 8.2.2 Annotating Erroneous Structures

User errors in writing can occur due to various reasons such as less care in writing or lower non-native language skills. These errors range from simple punctuation mistakes to completely ungrammatical structures which are incomprehensible. The followings are examples of both cases taken from the Foreebank:

(8) `This paragraph further confuses the issue?`

(9) `5. I was of cause a little bit ??!`

This section explains the strategy used to address the annotation of erroneous sentences. To the best of our knowledge, this annotation strategy is novel and the

main idea behind it is to mark the errors on the parse tree. Annotating the errors does not only make various analysis of the forum text possible (see Section 8.3), but also enables us to measure the effect of these errors on the parsing (see Section 8.5).

Prior to correcting the parse trees, the annotators are asked to *minimally* correct the user errors in the sentence itself.[11] In general, user errors can be categorized into the following types:

1. dropping required tokens
2. inserting extraneous tokens
3. substituting tokens with incorrect but valid ones (real word spelling errors)
4. spelling mistakes
5. arranging tokens in the wrong order

Once a sentence is corrected for these errors, the automatic parse tree of the *original* sentence is corrected with the aid of the corrected sentence based on the following guidelines:

- A **deleted** (missing) token is inserted into the tree with its correct POS tag suffixed with `_D`.
- An **extraneous** token is tagged with `Y_X`. The Y represents an unknown POS tag for such tokens.[12] The tag is moved to the lowest most appropriate level in the tree.
- A **real word spelling error** is tagged with the POS tag of the correct one suffixed with `_W` (standing for wrong word). The concept of real word errors ranges from very close word forms (e.g. *they* instead of *them*) to different parts of speech (e.g. *good* instead of *better*) or to completely a different meaning or usage (e.g. *desk* instead of *chair*).
- A **spelling error**, when a token is misspelled into exactly one token, is tagged with the `_S` suffix. A token is considered as a case of spelling error if both of

[11] Note that the word forms are kept intact in the trees.

[12] PTB and Web Treebank use X for unknown constituents, but do not envision any POS tags for such tokens. We introduce Y as a new POS tag for this type of tokens.

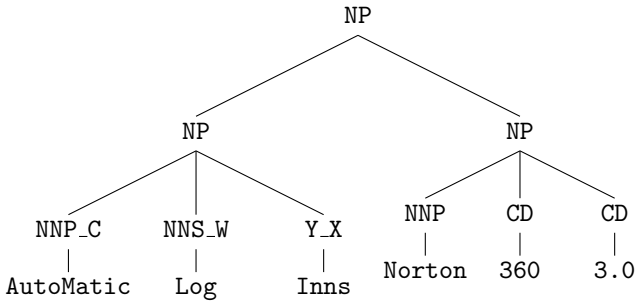



Figure 8.3: The Foreebank annotation of the sentence *AutoMatic Log Inns Norton 360 3.0* corrected as *Automatic Logins Norton 360 3.0*

the following conditions hold: 1) it is not what it was meant to be 2) it is not a valid token such as dictionary words, slang, internet or text messaging abbreviations, proper names, name initials, emoticons, etc.

- A **capitalization error** is treated as a special case of spelling error with the _C suffix. Examples are *norton internet security* instead of *Norton Internet Security* or *nis* instead of *NIS*.
- A **word order error** (at any distance) is treated as a combination of dropped and excess tokens at the highest possible level in the tree.
- All other errors which are not explicitly mentioned in this guidelines are tagged with the _E suffix.

These suffixes are only attached to the POS tags. When a whole constituent qualifies for one of these categories, the appropriate suffix is attached to all pre-terminals under it. It should be noted that, in real word (_W), spelling (_S) and capitalization (_C) errors, the surface form is not corrected on the tree, but only in the corrected sentence.[13] An example annotated sentence is presented in Figure 8.3. The original sentence is *AutoMatic Log Inns Norton 360 3.0* which is corrected as *Automatic Logins Norton 360 3.0*.

In addition to user errors, there are cases of innovative use of language occurring in this type of text. Initialisms such as *idk* standing for *I don't know* are examples of these cases. Such initialisms are neither corrected in the sentence nor broken down

[13]The reason is to avoid losing the original sentence form in the tree.

into multiple nodes in the tree, even though this can interfere with part of speech tagging and treebank annotation. Instead, their tag is suffixed with `_I`.

## 8.3 Analysing the Foreebank

In this section, we analyse the Foreebank text and its annotation by extracting statistics targeting various characteristics of the data set. Some of these statistics are compared to the WSJ and FTB for English and French respectively in order to better understand the difference between these data sets which can potentially affect the performance of parsers trained on these text types. We first analyse the textual specifications of the data. We then extract some statistics about the syntactic annotation of the Foreebank data sets. At the end, we measure the user error rate in the Foreebank text.

### 8.3.1 Characteristics of the Foreebank Text

Table 8.1 presents some characteristics of the English and French Foreebank text such as the average sentence length and compare them to those of the WSJ and FTB corpora. It also gives the out-of-vocabulary (OOV) rate of these data sets with respect to the WSJ and FTB. According to the table, the Foreebank sentences are shorter in average than the WSJ and FTB sentences. While the standard deviation of the English data sets are similar, the standard deviation of the FTB data set is higher than that of the French Foreebank due to the existence of very long sentences in the FTB. The table also shows that the OOV rate of both English and French Foreebanks with respect to their corresponding edited new treebanks are high. These numbers can be compared to the OOV rate of the WSJ test section with respect to its training section which is 13.2% and the FTB test section with respect to its training section which is 21.6%. It can be seen that the OOV rate of the French Foreebank is higher than the English one, most probably due to the larger size of the WSJ compared to the FTB. Additionally, the OOV rate of the English Foreebank

Table 8.1: Characteristics of the English and French Foreebank corpora compared with those of the WSJ and FTB. The OOV rates are computed with respect to WSJ and FTB for the English and French Foreebank respectively.

| | English | | French | |
|---|---|---|---|---|
| | Foreebank | WSJ | Foreebank | FTB |
| Average sentence length | 15.4 | 23.8 | 19.6 | 28.4 |
| Sentence length SD | 11.1 | 11.2 | 12.8 | 16.5 |
| Maximum sentence length | 89 | 141 | 86 | 260 |
| OOV rate | 33.3% | - | 39.1% | - |

is more than 2.5 times as big as that of the WSJ test set, while the OOV rate of the French Foreebank is less than 2 times as big as that of the FTB test set. This suggests that a bigger performance drop due to unknown words should be expected in parsing the English Foreebank than the French Foreebank. This will be further clarified in Section 8.4, where the parsers are evaluated on the Foreebanks.

### 8.3.2 Characteristics of the Foreebank Annotations

Table 8.2 displays the number of each specific tag suffix explained in Section 8.2 in the annotation of the English and French Foreebank and their percentage with respect to the total number of tokens.[14] These suffixes represent the errors made by the user. The statistics for the French Foreebank are presented separately for the monolingual (`mono`) and translated (`trans`) sections. According to the table, the capitalisation error is the most frequently occurring error for both languages, mainly due to the existence of many product names in the corpora. While deleted tokens are as frequent as the capitalisation errors in the English data set. The are not as frequent in the French data sets, especially in the translated section. Spelling errors are the next most frequent problem, albeit mainly in the monolingual section in the case of French Foreebank. Real word errors occur as often as the spelling errors in the English Foreebank. They are also as frequent in the French Foreebank and

[14]The percentage of the merged sentence suffix (`_M`) is computed with respect to the number of sentences instead of tokens.

Table 8.2: Number and percentage of tag suffixes in the English and French Foreebank annotation

| Suffix | Explanation | English | | French | | | |
|---|---|---|---|---|---|---|---|
| | | | | mono | | trans | |
| | | # | % | # | % | # | % |
| _M | Merged sentences | 3 | 0.3% | 3 | 0.6% | 24 | 4.8% |
| _D | Deleted token | 120 | 0.16% | 32 | 0.07% | 7 | 0.01% |
| _X | Extraneous token | 33 | 0.04% | 4 | 0.01% | 4 | 0.01% |
| _W | Real word error | 70 | 0.09% | 34 | 0.08% | 3 | 0.01% |
| _S | Misspelled token | 76 | 0.1% | 101 | 0.25% | 9 | 0.02% |
| _C | Capitalisation error | 124 | 0.16% | 110 | 0.26% | 72 | 0.12% |
| _B | Broken token | 1 | 0% | 13 | 0.03% | 6 | 0.01% |
| _I | Innovative initialism | 1 | 0% | 8 | 0.02% | 0 | 0% |
| _E | Other errors | 1 | 0% | 1 | 0% | 0 | 0% |

the next most occurring errors after spelling mistakes in the monolingual section of this data set. It can also be seen that there are more cases of extraneous tokens in the English than in the French Foreebank. Finally, merged sentences are found to be frequent in the translated section of the French Foreebank, mainly due to the original sentence segmentation of this data.

We additionally find that the most frequent POS tags carrying user errors in the English Foreebank are NNP (proper nouns) with capitalization errors and , (for , and - and /) as deleted token. For the French, they are NC (common noun) and NPP (proper noun) with capitalization errors.

In sum, it seems that capitalization of the proper nouns is the major error in the Foreebank data set, especially in the French one, mainly due to product names. Deleted tokens are also a major source of problem in the English Foreebank. Overall, the errors occur on only a small fraction of the tokens in both data sets. The next section provides further analysis of these errors.

### 8.3.3 User Error Rate

In order to find the level of user error in the forum text, we calculate the edit distance between the original sentences and their edited versions using the special suffix tags

Table 8.3: Number of user errors and the edit distance between the original and edited Foreebank sentences; Ins: inserted (extraneous), Del: deleted (missing), Sub: substituted, Total: Ins+Del+Sub, anED: average normalised edit distance

| | Ins | Del | Sub | Total | anED |
|---|---|---|---|---|---|
| English Foreebank | 33 | 120 | 270 | 423 | 0.03 |
| French Foreebank `mono` | 4 | 32 | 245 | 281 | 0.04 |
| French Foreebank `trans` | 4 | 7 | 84 | 95 | 0.01 |

on the POS tags of the tokens in the Foreebank annotation. We consider three error categories: 1) inserted (extraneous) tokens identified by the `_X` tag suffix, 2) deleted (missing) tokens identified by the `_D` tag suffix, and 3) substituted tokens including spelling errors (`_S` tag suffix), real word errors (`_W` tag suffix) and capitalization errors (`_C` tag suffix). The number of these suffixes is counted for each sentence and the edit distance is computed by summing them and normalised by dividing the sum by the maximum of the lengths of the original sentence and its edited version.

Table 8.3 shows the results for both English and French Foreebank, with the latter broken down to the monolingual (`mono`) and translated (`trans`) sections. The total number of insertions (Ins), deletions (Del) and substitutions (Sub) as well as their sum and the average normalised edit distance at the document level are given in the table. The results reveal that, despite the existence of some near to incomprehensible erroneous sentences, the overall error level is very low. As also observed in the previous section, there are more extraneous and missing words in the English Foreebank than in the French ones. However, the numbers of substitutions are closer (270 vs. 245+84) and bigger for the French. Not surprisingly, the translated sentences of the French Foreebank contain fewer errors.[15] In Section 8.5, we will investigate the part these errors play in the parsing performance of the forum text.

[15]The reason for these errors existing in the translated sentences in the first place is that the translation guidelines emphasise the minimal transformation of the source sentences during the translation, noting that most of the substitution errors (72 out of 84) are the capitalization problems.

## 8.4 Parser Performance on the Foreebank

With the availability of Foreebank, it is now possible to evaluate the performance of the parsers applied to the forum text in the previous experiments. Comparing these performances to those on the in-domain evaluation sets will help understand the amount of loss due to the domain shift and text type change from newswire to Norton user forums. To this end, we parse the English and French Foreebanks using the Lorg parsing models used to parse the SymForum data set for the QE experiments. The English model is trained on the entire WSJ and the French model on the entire FTB. However, we also need to evaluate the parser performance on the in-domain test data of the parsing models. Therefore, we additionally parse the Foreebank as well as the test sections of WSJ and FTB using the Lorg parsing models trained only on the training sections of the corresponding treebanks.

As explained in the previous section, the Foreebank trees contain annotations for the user errors with special suffixes on the labels and insertion of missing words (D-suffixed nodes). Since these suffixes are not present in the WSJ and FTB, we remove them for these experiments. We also remove the D-suffixed nodes and their corresponding tokens since we parse the original sentences.

Table 8.4 and Table 8.5 display the evaluation results of these parsing models on the English and French Foreebank sentences as well as the test sets of the WSJ and FTB respectively. The results for French Foreebank are presented separately for each section. There is a large $F_1$ gap of about 15 points between the parsing accuracies of the English Foreebank and WSJ test set. The gap is substantially bigger than what has been reported for between the WSJ and other corpora such as the Brown and British National Corpus (Foster and van Genabith, 2008). This is probably due to a greater distance between the WSJ and the Foreebank than between the WSJ and those corpora in terms of vocabulary and the grammatical constructions, as we found in Section 8.3.3 that only a small amount of editing was required to correct the user errors in the Foreebank. In addition, this gap is also bigger that between

Table 8.4: Comparison of parsing WSJ and forum text using a WSJ-trained parser

| Test set | Training set | P | R | $F_1$ |
|---|---|---|---|---|
| Foreebank | WSJ Whole | 76.10 | 77.00 | 76.55 |
| Foreebank | WSJ Train | 74.21 | 75.61 | 74.90 |
| WSJ test | WSJ Train | 89.95 | 89.15 | 89.55 |

Table 8.5: Comparison of parsing FTB and forum text using a FTB-trained parser

| Test set | Training set | P | R | $F_1$ |
|---|---|---|---|---|
| Foreebank `mono` | FTB Whole | 73.84 | 75.15 | 74.49 |
| Foreebank `mono` | FTB Train | 73.83 | 74.61 | 74.22 |
| Foreebank `trans` | FTB Whole | 78.65 | 78.9 | 78.77 |
| Foreebank `trans` | FTB Train | 78.56 | 78.2 | 78.38 |
| FTB Test | FTB Train | 83.53 | 83.28 | 83.40 |

the WSJ and English Web Treebank parsing reported by Petrov and McDonald (2012) as well as parsing WSJ and the sports discussion forum reported by Foster et al. (2011a) (both by about 4 points). However, it is smaller than the 19 $F_1$ points difference between parsing WSJ and the tweets observed by Foster et al. (2011a), which is expected due to a farther distance between the WSJ and Twitter in terms of sentence structure and also vocabulary. In Section 8.5, we will try to shed more light on this matter.

The performances of parsing the English Foreebank and the monolingual section of the French Foreebank are similar when the English parser is trained on the training section of the WSJ (74.90 vs. 74.49 and 74.22 $F_1$ points). However, compared to parsing the English Foreebank, the performance drop in parsing French Foreebank is relatively smaller: the former drops from 89.55 $F_1$ points to 74.90 and the latter from 83.40 to 74.22 and 78.77 for the monolingual and translation sections respectively. This suggests that the French parsing model is better generalisable to the forum text, or alternatively, the FTB test set is more distant from its training set than the WSJ one, which confirms our anticipation based on the OOV rate in Section 8.3.1. The difference with parsing the FTB test set is just above 9 $F_1$ points for the

monolingual section and 5 points – even smaller – for the translated section. This 4 points difference may be due to the higher level of user error in the monolingual section. This will be further clarified in Section 8.5.

The effect of using the combined training and evaluation sections of the WSJ and FTB instead of only their training sections is also worth noting. While adding the WSJ development and test sets (about 5,500 sentences) increases the $F_1$ score of the Foreebank parsing by 1.5 points, the 2,500 FTB development and test sentences have a little effect on parsing the French Foreebank. In the former case, the training size increases by approximately 14%, whereas in the latter case the increase is 25%. Therefore, contrary to this result, the supplementary training data is normally expected to be more useful for the FTB which is also much smaller than the WSJ and should benefit more from the additional training data. This suggest that the new FTB sentences are still not enough or do not bring additional information to the parsing model. Comparing to the results reported by Gildea (2001), where adding additional training data from a different domain than the test set had no benefit, we can see that additional WSJ evaluation sections, which is even smaller than the Brown, improve parsing of the Foreebank. On the other hand, similar to the result of Gildea (2001), additional in-domain training data is not effective for the French parsing, although the amounts of additional data are not comparable.

## 8.5 The Effect of User Error on Parsing

As explained in Section 8.2.2, as part of the Foreebank annotation, we ask the annotator to minimally correct the errors made by the forum users. This provides an opportunity to examine the effect of these errors on the parsing accuracy. We therefore parse the corrected versions of the sentences in both English and French Foreebank and compare their accuracy measured by the PARSEVAL metric with the accuracy of their original version. For parsing the sentences, we use the parsing models described in Section 4.3 of Chapter 4 used to parse the SymForum data set.

Table 8.6: Comparison of parsing original and edited forum sentences

| | English | | | French | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | mono | | | trans | | |
| | P | R | $F_1$ | P | R | $F_1$ | P | R | $F_1$ |
| Original | 76.10 | 77.00 | 76.55 | 73.84 | 75.15 | 74.49 | 78.65 | 78.90 | 78.77 |
| Edited | 77.62 | 78.61 | 78.11 | 74.69 | 76.03 | 75.35 | 78.84 | 79.14 | 78.99 |

For English, it is the Lorg parser trained on the entire WSJ and for French the same parser trained on the whole FTB.

The performances of the parsers are shown in Table 8.6. For French, the evaluation on the monolingual (`mono`) and translated sections (`trans`) of the Foreebank are presented separately. According to the results, correcting the user errors before parsing leads to an improved parsing accuracy for English Foreebank, where the new parses achieve more than 1.5 points higher $F_1$, both precision and recall increasing similarly. Considering the amount of edits on each data set shown in Table 8.3, this improvement is noticeable. On the other hand, despite a (slightly) higher edit distance for the monolingual French Foreebank (0.4 vs. 0.3), the smaller impact is observed on the monolingual section of the French Froeebank (about 1 $F_1$ point). Parsing the translated section has not changed. Noting to the detailed error corrections in Table 8.3 suggests that the inserted and deleted tokens may have a larger effect on parser errors than the substituted tokens, as their number is higher for the English. This can be explained by considering that the correction of substitution errors normally has a small effect on the phrase structure. For example, correcting the capitalisation errors mainly changes the `NN` tags to `NNP`, which is not expected to affect the tree structure as much as the insertion or deletion of a node. This is in contrast to the results obtained by Foster et al. (2008), where the real word spelling errors has the largest effect on the parsing of the artificially generated noisy WSJ. However, it is on par with the observation by Foster (2010), where the effect of spelling errors is found to be small on parsing the sports discussion forum, a more similar text to the Foreebank than the WSJ.

Table 8.7: Using English Foreebank as training data, both alone and as a supplement to the WSJ, and also supplemented by the English Web Treebank (EWT) evaluated in a 5-fold cross validation setting on the Foreebank

| Training set | Lorg Constituency | | | Stanford Dependency | |
|---|---|---|---|---|---|
| | P | R | $F_1$ | LAS | UAS |
| WSJ (Whole) | 76.06 | 76.95 | 76.50 | 74.26 | 79.41 |
| Foreebank | 69.61 | 69.27 | 69.44 | 64.55 | 71.63 |
| WSJ+Foreebank | 78.01 | 78.71 | 78.36 | 76.51 | 81.14 |
| WSJ+Foreebank+EWT | 77.96 | 79.01 | 78.48 | 76.67 | 81.34 |

## 8.6 Improving Parsing of Forum Text

The simplest method to improve the accuracy of parsing forum text by exploiting the Foreebank is to use it as a supplementary training data set to the WSJ and FTB. In this section, we combine the two treebanks and retrain the English and French parsers used to parse the SymForum data for syntax- and semantic-based quality estimation in the previous chapters. These parsers include the English and French Lorg constituency parsers which are already evaluated on the Foreebank in Section 8.4 and additionally the dependency conversions of the English parses by the Stanford converter and the French dependency parses by the ISBN parser.[16] In order to test the parsers, we run a 5-fold cross validation, in which the Foreebank is randomly split into five parts, each used for the evaluation of the parsers trained at each fold on either of the WSJ or FTB plus the other four parts. The entire WSJ and FTB are used. Additionally, we evaluate the parsing models built using the English and French Foreebank alone as well as the WSJ and FTB alone within the same cross validation setting.

Table 8.7 and Table 8.8 show the results for English and French respectively. Note that, to have bigger test sets at each fold, the French Foreebank is used as a whole instead of splitting it into its two subsections. According to Table 8.7, the parser trained on the Foreebank is substantially outperformed by the parser

[16]The ISBN model is trained on the dependency conversion of the French Foreebank using the `Const2Dep` tool.

Table 8.8: Using French Foreebank as training data, both alone and as a supplement to the FTB, evaluated in a 5-fold cross validation setting on the Foreebank

| Training set | Lorg Constituency | | | ISBN Dependency | |
|---|---|---|---|---|---|
| | P | R | $F_1$ | LAS | UAS |
| FTB (Whole) | 76.62 | 77.33 | 76.97 | 72.29 | 79.42 |
| Foreebank | 72.46 | 72.35 | 72.40 | 76.36 | 80.10 |
| FTB+Foreebank | 79.59 | 80.36 | 79.98 | 76.04 | 82.31 |

trained on the WSJ, showing that a small amount of in-domain training data is not preferable to a much larger amount of out-of-domain data. This applies to both constituency and dependency parsing. Combining the WSJ and the Foreebank improves the $F_1$ of the constituency parsing by about 2 points over when only the WSJ is used for training. The dependency parsing also benefits from this combination to a similar extent (above 2 points of LAS). Considering that Foreebank is orders of magnitude smaller than the WSJ, the gain by its addition to the WSJ is noticeable.

On the French side, as seen in Table 8.8, we observe slightly bigger improvements by combining the FTB and Foreebank, where the $F_1$ of constituency parsing is increased by 3 points and the LAS of dependency parsing by approximately 4 points. However, we observe a different behaviour using only Foreebank for training. First, there is a smaller gap of 3.5 $F_1$ points between the performance of the constituency parsing model trained in this way and the one trained on the FTB, compared to 7 points for English. This is probably due to the fact that the proportional size of the French Foreebank with regard to the FTB is bigger than that of the English Foreebank with respect to the WSJ. Second, the dependency parses of the ISBN model trained on the Foreebank achieve higher scores than the one trained on the FTB, especially in terms of LAS which is even slightly higher than that of the combined model. This is in contrast with the results of English dependency parsing in Table 8.7. This difference may be related to the way the French and the English dependency parses are obtained. The French dependency parses are the results of parsing with a model trained on the conversions of the gold-standard Foreebank,

unlike the English ones which are converted from the automatic constituency parses. Therefore, the English dependency parses are probably affected by the low quality of their source constituency parses, while the French dependency parser has been able to better exploit the less noisy dependency structures.[17]

The size of the supplementary Foreebank data used above is very small. It is expected that a larger amount of such data will further improve the parsing performance. Since additional hand-annotated parse trees from the Norton forum text are not available, it is interesting to know if other data which have some similar characteristics in common with this data can be used as a replacement. For example, although not in the same domain, the English Web Treebank is similar to our data in that it is user-generated, extracted from the Web and contains similar text types such as newsgroups, *Yahoo!* answers and emails. We therefore experiment with adding this data set to the combination of the WSJ and Foreebank and retraining the English parsers. The evaluation results are shown in the last row of Table 8.7. According the results, despite its relatively large size (about 15K sentences), there are only tiny increases in the scores using the English Web Treebank as additional training data. This suggests that the similarity in the vocabulary is more important than the similarity in the style of the text. It is worth noting that the precision of the constituency parsing is slightly lower with this parsing model, a measure which is perhaps more important in the quality of downstream semantic role labelling.

Another way to compensate for the small proportion of the Foreebank used to supplement parser training data is to increase the weights of the parsing rules extracted from those trees in the grammar. To accomplish this goal, we simply replicate the Foreebank trees in the training sets of both the English and French parsers. For the English parser, we start with experimenting two settings: one contains 5 times replication of the Foreebank and the other 10 times. For the

[17]Although the dependency trees on which the ISBN is trained are the conversions of the gold-standard Foreebank, the conversion process involves a statistical process which automatically labels the trees with function tags required by the converter. This may introduce a little noise to the resulting dependency trees.

Table 8.9: Replicating the English Foreebank in the training data of the parsers (highest scores in bold)

| Training set | Lorg Constituency | | | Stanford Dependency | |
|---|---|---|---|---|---|
| | P | R | $F_1$ | LAS | UAS |
| WSJ+Foreebank | 78.01 | 78.71 | 78.36 | 76.51 | **81.14** |
| WSJ+Foreebank×5 | **78.89** | **79.27** | **79.08** | 76.51 | 80.78 |
| WSJ+Foreebank×10 | 77.93 | 78.95 | 78.44 | **76.69** | 81.08 |

Table 8.10: Replicating the French Foreebank in the training data of the parsers (highest scores in bold)

| Training set | Lorg Constituency | | | ISBN Dependency | |
|---|---|---|---|---|---|
| | P | R | $F_1$ | LAS | UAS |
| FTB+Foreebank | 79.59 | 80.36 | 79.98 | 76.04 | 82.31 |
| FTB+Foreebank×3 | **80.02** | **80.74** | **80.38** | **78.16** | **82.97** |
| FTB+Foreebank×5 | 79.95 | 80.65 | 80.30 | 78.01 | 82.77 |

French, the in-domain portion of which is smaller, we also experiment two settings but one containing 3 and the other 5 times replications of the Foreebank. The goal is to increase the number of replications if the higher number of replicates performs better in these settings.

Table 8.9 displays the results of the replications for English and Table 8.10 for French. For comparison, the scores for one replications are repeated from Table 8.7 and Table 8.8 respectively (in grey). Looking at the results for English first, it can be seen that replicating the Foreebank five times (WSJ+Foreebank×5) helps improve the constituency parses over when only one copy of it is used in the combination, although the gain is small. However, the quality of the dependency conversions of these parses does not improve. On the other hand, further replication of the Foreebank seems to adversely affect the parsing rules extracted from its trees, as the evaluation scores of the constituency parsing model trained on WSJ+Foreebank×10 suggest. The dependency conversion of these parses is also of a similar quality to the other models, as indicated by their scores.

The replication of the Foreebank is also useful for parsing the French Foreebank as shown in Table 8.10 for French. However, in contrast to the English, the dependency parses benefit more than the constituency parses. On the other hand, the effect of further replication is similar to what was observed for English, i.e. slightly detrimental.

In sum, adding in-domain data to the parser training set improves the parsing performance, even though only a small amount compared to the exiting out-of-domain data is added. However, the larger out-of-domain training set cannot be replaced completely. Replicating this small amount in the combination can be a further help albeit not significantly. In the next section, we apply the new parsers to the SymForum data and use the resulting parses in semantic role labelling of the data which is subsequently used to rebuild the semantic-based QE systems of the previous chapter.

## 8.7 Semantic-based QE with Improved Parses

The objective of improving the performance of parsing of the forum text was to improve the accuracy of the quality estimation of machine translation of this type of text. In Chapter 5, we found that the parsing performance did not affect the accuracy of syntax-based quality estimation. However, as we discussed earlier in this chapter, the performance of parsing can influence the accuracy of semantic-based QE through its impact on the quality of underlying semantic role labelling. Inspired by this idea, we acquire new semantic role labellings of the SymForum data using the improved parses from the previous section. Specifically, the English side of the SymForum is parsed using the Lorg parser trained on the entire WSJ plus five replications of the Foreebank and converted to dependencies (the model in the second row of Table 8.9). The new constituency and dependency parsing models achieve 2.5 $F_1$ and 2.3 LAS points above the original models respectively according to Tables 8.7 and 8.9. The French side of the data set is parsed using the Lorg parser trained on

the entire FTB plus three replications of the Foreebank to obtain the constituency parses and using the ISBN parser trained on the dependency conversions of these parses. When measured in terms of LAS, the new English dependency parses are 80% and the French ones 73% similar to those used in Chapter 7 for SRL.

Once the data is parsed, they are semantic role labelled using the same models used in Chapter 7. The new parsing models perform 3.5 $F_1$ and 6 LAS points higher for constituency and dependency parsing respectively as can be seen in Tables 8.8 and 8.10. When measured in terms of $F_1$, the new English semantic role labelling is 85% and the French one 76% to the labelling obtained in Chapter 7. The semantic-based QE systems are then built using these labellings. From among the semantic-based QE systems built in Chapter 7, we replicate the word alignment-based PAM (`WAPAM`) and the combined semantic-based QE system (`SeQE`) as presented in the next sections.

### 8.7.1 The Word Alignment-based PAM

In Section 7.5.1 of Chapter 7, we introduced the `WAPAM` metric to estimate the translation quality by matching the semantic predicate-argument structure of the source and target using the word alignments between them. However, we found in Section 7.5.4 of the same chapter that the low quality of the predicate-argument structure hindered the performance of this metric. We now examine this metric using the new semantic role labelling built upon the improved parses. From among the six different `WAPAM` score types, we choose the two best performing ones, namely unlabelled recall (`URec`) and $F_1$ ($\texttt{UF}_1$) instead of all scores.

Table 8.11 displays the results of the estimation with `WAPAM` using the new semantic role labelling of the SymForum data as well as the results of the old WAPAM from Chapter 7 (in grey) for comparison. The new metric scores are identified by the *new* subscript. According to the table, the estimations of all evaluation metrics improve using the new semantic role labelling except the HTER in terms of both measures and the Adequacy in terms of RMSE. The only statistically significant

Table 8.11: Performance of word alignment-based PAM (WAPAM) using the old (in grey) and new SRLs

| | 1-HTER | | HBLEU | | Adequacy | | Fluency | |
|---|---|---|---|---|---|---|---|---|
| | RSME | r | RSME | r | RSME | r | RSME | r |
| URec | **0.3249** | **0.2483** | 0.3706 | 0.2338 | 0.3213 | 0.2796 | 0.4171 | 0.2927 |
| URec$_{new}$ | 0.3321 | 0.2436 | **0.3675** | **0.2433** | **0.3198** | **0.2981** | **0.4041** | **0.3257** |
| UF$_1$ | **0.3175** | **0.2328** | 0.3607 | 0.2179 | **0.3108** | 0.2698 | 0.4033 | 0.2865 |
| UF$_{1new}$ | 0.3237 | 0.2289 | **0.3597** | **0.2278** | 0.3118 | **0.2806** | **0.3970** | **0.3082** |

change is the improvement of the Fluency estimation using unlabelled recall (URec). Considering the amount of improvement expected for the semantic role labelling of the data based on the amount of improvement we observed in the previous section for their parsing quality, the gains on the WAPAM scores seem to be encouraging. However, this metric relies on the balance between the quality of semantic role labelling of the source and target in addition to their absolute quality, since it can interpret a predicate or argument missed by the SRL system on either side as a missing or spurious translation. Therefore, the unexpected degradations can be attributed to the negligence of this balance here.[18]

### 8.7.2 The Semantic-based QE System

In addition to the PAM metric, in Section 7.7, we built SeQE, a statistical system which exploits the semantic role labelling of the source and target via tree kernels and hand-crafted features. This system is rebuilt here using the new semantic role labelling of the SymForum data.

Table 8.12 shows the performance of this system (SeQE$_n ew$), as well as the SeQE replicated from the previous chapter (in grey). All the scores have increased using the new SRL. The changes for the human-targeted metrics are especially bigger. The HTER prediction achieves the highest improvements and the Fluency prediction

[18]It should be noted that the WAPAM scores used here are the unlabelled scores. As we observed in Section 7.2 of Chapter 7, English and French SRL achieve very close unlabelled scores, when evaluated on the in-domain data. However, the SRL models are applied to the MT output in the French side, which can also result in further noise. Moreover, these in-domain evaluation data are different for the two languages and the comparison is not completely meaningful.

Table 8.12: Semantic-based QE systems using the old (SeQE) and new SRLs ($\texttt{SeQE}_{new}$); underlined scores are statistically significantly better.

| | 1-HTER | | HBLEU | | Adequacy | | Fluency | |
|---|---|---|---|---|---|---|---|---|
| | RSME | r | RSME | r | RSME | r | RSME | r |
| SeQE | 0.2249 | 0.3884 | 0.2710 | 0.3648 | 0.2242 | 0.4447 | 0.2404 | 0.5182 |
| $\texttt{SeQE}_{new}$ | **0.2214** | **0.4223** | **0.2680** | **0.3892** | **0.2224** | **0.4584** | **0.2390** | **0.5233** |

the lowest. However, only the HTER prediction changes are statistically significant. When the system is broken down into its tree kernel and hand-crafted component, we find that the improvements are contributed by the tree kernels. This can be seen in Table 8.13, where the performance of the tree kernel component using the new SRL ($\texttt{SSTK}_{new}$) is higher than the one built in Section 7.3 of Chapter 7 (SSTK) but the hand-crafted system using the new SRL ($\texttt{SeHC}_{new}$) performs significantly worse than the system built using the old SRL (SeHC[19]). This suggests that the tree kernels are able to make use of the improved semantic role labelling as they have a better means of exploiting them via having access to the entire annotated tree. While the improvements of the tree kernel system are only statistically significant for the prediction of the human-targeted metrics, the hand-crafted system using the new SRL degrades statistically significantly for manual metric prediction. Moreover, the magnitude of the degradations in the case of hand-crafted system tends to be bigger than the extent of the improvement seen for the tree kernels systems. This renders the combination of these two components successful as the improved component seems to outweigh the degraded one so that the combined system ($\texttt{SeQE}_{new}$) also improves using the new SRL and is also better than the $\texttt{SSTK}_{new}$.

All in all, it seems that a more significant improvement is required in the performance of the SRL of the SymForum data in order to be sufficiently effective in the downstream semantic-based QE. Additionally, we conjecture that a balanced semantic role labelling quality between the source and target, where the French SRL especially when applied to the MT output, in our case, has been improved to the

[19]This system was named $\texttt{SeHC}_{+pam}$ in Section 7.5.5 of Chapter 7.

Table 8.13: Semantic tree kernel and hand-crafted QE systems using the old (in grey) and new SRLs (subscripted with *new*); underlined scores are statistically significantly better.

| | 1-HTER | | HBLEU | | Adequacy | | Fluency | |
|---|---|---|---|---|---|---|---|---|
| | RSME | r | RSME | r | RSME | r | RSME | r |
| SSTK | 0.2269 | 0.3682 | 0.2722 | 0.3537 | 0.2253 | 0.4351 | 0.2425 | 0.5046 |
| $\text{SSTK}_{new}$ | **0.2224** | **0.4152** | **0.2692** | **0.3788** | **0.2246** | **0.4409** | **0.2404** | **0.5158** |
| SeHC | **0.2445** | **0.2387** | **0.2822** | **0.2368** | **0.2370** | **0.3571** | **0.2575** | **0.3908** |
| $\text{SeHC}_{new}$ | 0.2449 | 0.2278 | 0.2873 | 0.2083 | 0.2459 | 0.3077 | 0.2651 | 0.3550 |

same level as the English SRL, can boost the performance of semantic-based quality estimation both using the PAM metric and the statistical systems.

## 8.8 Summary and Conclusion

In this chapter, we developed two constituency syntax treebanks, one for English and one for French, from Symantec Norton forum text. The aim of creating such resources was to use them in the evaluation and improvement of parsing Norton forum text, with a view to improving QE. These parses have been used in the quality estimation systems built throughout this thesis both directly in the syntax-based QE systems and indirectly in the semantic-based QE systems via semantic role labelling systems built upon them. We devised new annotation strategies to handle phenomena specific to user-generated content. Instead of correcting user errors prior to annotation, we addressed them during the annotation of the parse trees themselves via a set of suffixes attached to the POS tags of the erroneous tokens. This does not only help account for real-word scenarios where human intervention to correct such errors prior to parsing is not possible, but also provides a means of annotating user errors and measuring the user error rate in the data.

Using these data sets, we found that only a small fraction of the tokens contain any kind of user errors, most of which are spelling and capitalization errors. Correcting these errors could improve the parsing performance, to a relatively noticeable

degree considering their small quantity.

We also used these data sets to supplement the training data of the original parsers and found that it could boost the performance of parsing the Norton forum text, despite the small sizes of these supplementary data sets. Replicating these data sets in the training sets to increase their weights in the resulting parsing models proved to be slightly useful.

The improved parsers are finally applied to the SymForum data set used in the quality estimation experiments in the previous chapters with the aim of improving the semantic role labelling of this data. The PAM QE metric and the semantic-based QE system are then rebuilt using the new labellings. While the accuracy of the PAM and the semantic-based system in estimating most of the evaluation metrics improved, only some of the changes were statistically significant. We conclude that further enhancement of the parsing of the forum text, for which there is a considerable room, can increase the performance of the semantic-based quality estimation by improving the underlying semantic role labelling. However, the lower quality of French semantic role labelling, especially on the MT output, causing the imbalance between the source and target labelling still remains a crucial stumbling block.

# Chapter 9

# Conclusion and Future Work

In this thesis, we explored the use of syntactic information in machine translation of user-generated content and both syntactic and semantic information in its quality estimation. The user-generated content used in this thesis is from Symantec's online Norton forums which contain discussions of Norton security products posted by Norton users and employees. Since the majority of the content is in English, the use of machine translation is needed to disseminate this content in other languages, so that a wider range of users can avail of the information and knowledge contained therein. However, a measure of confidence is required to assure that only high quality and legitimate translations are published. Therefore, a reliable estimation of translation quality is as crucial as a high quality translation.

Much research has been dedicated to incorporating syntactic knowledge in the statistical machine translation process to account for problems such as long distance word order which cannot be tackled by methods merely modelling the translation of words or sequences of words called (ad-hoc) phrases. We analysed the output of these two spectra of machine translation methods, i.e. syntax-based and phrase-based, to find out whether the currently used syntax-based methods are able to better handle such problems. We also compared these methods to discover a systematic difference between them to be utilized in a combination framework.

The quality of translation can be judged by its fluency and adequacy. While flu-

ency is related to its grammaticality, adequacy is concerned with the match between the semantics and pragmatics of the translation and its source. We investigated methods of using syntactic and semantic knowledge in estimating the translation quality via in-depth analyses. The semantic-based quality estimation involved attempts to improve the semantic role labelling of French due to the lack of sufficient SRL resources for this language.

The syntactic and semantic analyses used in the quality estimation experiments in this thesis were obtained using tools built upon resources created from edited newswire text. To evaluate the performance drop due to this shift in domain and text type, we built two treebanks, one for English and one for French, taken from the Norton forum text and annotated using an approach tuned to address the phenomena specific to such unedited text. This annotation strategy enabled us to analyse the user errors present in this type of text. We additionally used these treebanks to supplement the training data of our parsers. The improved parses were ultimately used to replicate our semantic-based QE systems.

In the next section, we summarise the experiments carried out in each chapter to answer our related research questions and discuss the findings. Section 9.2 lists the contribution of the thesis. Finally, the last section discusses directions for future work.

## 9.1 Summary and Findings

In Chapter 2, we compared phrase-based, hierarchical phrase-based and syntax-based methods in translating Symantec translation memory content as well as user-generated Norton forum text. The first research question we addressed is as follows:

*How different are the outputs generated by each of these methods?*

To answer this question, was automatically compared the output of the translation systems built using these methods with a variety of widely used MT evaluation metrics at the sentence and document level and on both 1-best and n-best results.

We found that different systems tend to generate different outputs for the same sentences, especially in more difficult translation tasks such as out-of-domain translation. Based on these differences, we asked the following research question:

*Can the outputs be beneficially combined in theory?*

Using sentence-level oracle combination, we found a system selection on both 1-best and 500-best outputs produced by these systems can lead to significant improvement. The gains by oracle combination were especially larger for the 500-best outputs. Based on this results, we posed the third research question as follows:

*Are any differences between the two types of systems systematic enough to be exploited in system combination?*

In order to answer this question, we manually compared the hierarchical phrase-based and the string-to-tree syntax-based methods in terms of various lexical and grammatical translation phenomena. We found that the syntax-based methods did not perform particularly better in handling grammatical problems such as long-distance reordering despite what is generally assumed. The lack of a systematic pattern in the differences between these methods and their output renders it difficult to develop a framework in which such differences can be exploited. We conjecture that the relaxation method used to loosen the syntactic constraints imposed during translation rule extraction to broaden the rule coverage (without which the performance of the syntax-based methods is considerably low) blurs the boundaries between ad-hoc and syntactic phrases, and, consequently, between the phrase-based and syntax-based methods.

In Chapter 3, we introduced SymForum, a data set we built for the experiments on quality estimation of machine translation of Norton forum text, containing the machine translation of English forum sentences to French, their human post-edits as well as adequacy and fluency scores. Analysing this data set showed that there is a high correlation between adequacy and fluency scores and even higher correlation between the human-targeted metric scores. We additionally created another data

set with sentences selected from the same domain as the parsers' training data, i.e. newswire. The purpose of this data set was to further validate the effectiveness of our syntax-based QE methods in the absence of noise resulting from out-of-domain application of the parsers to Norton forum text, which could influence the conclusions.

Using these two data sets, we built and experimented with syntax-based QE systems in Chapter 4. The first research question we tackled is as follows:

> *How effective is syntactic information in quality estimation of machine translation both in comparison and in combination with other surface-driven features?*

To find the answer, we experimented with two different methods of encoding syntactic information derived from both constituency and dependency parses of the source and target, namely tree kernels and hand-crafted features. The tree kernel systems performed better than the hand-crafted features, eliminating the time required for engineering such features. We built a full syntax-based quality estimation system by combining the tree kernels and hand-crafted features. This system was able to outperform the baseline used in the recent WMT QE shared tasks. This system was also successfully combined with the baseline. The syntax-based systems showed a better performance on the News data set than the SymForum data set, however – possibly because this data is well-formed and more homogeneous leading to more consistent parses across the data set.

We investigates analysed in detail the role of syntax in quality estimation in Chapter 5. The first research question we asked ia as follows:

> *Does parsing accuracy affect the performance of syntax-based QE?*

Two answer this question, we rebuilt the syntax-based QE systems using parse trees produced by parsers trained on only a fraction of the data used by the original parsers leading to much lower parsing accuracies measured by PARSEVAL $F_1$. Interestingly, the new QE systems performed at the same level as the systems built

using the original high-accuracy parses, showing that parsing accuracy does not affect the syntax-based QE performance.

In addition, by teasing apart the roles played by the source and target syntax, we sought the answer to the following research question:

> *To what extent do the source and target syntax each contribute to the syntax-based QE performance?*

We found that French constituency parses of the target were considerably less useful than the English ones in the source side. This observation raised the following research question:

> *Does parsing of noisy machine translation output affect the performance of syntax-based quality QE?*

To find the answer, we reversed the translation direction under a similar setting and replicated the experiments. The results showed that the inferiority of the French parse trees was not due to the fact that they were the parses of potentially ill-formed machine translation output. We hypothesised that the flatter structure of the French Treebank trees used to train French parsers was responsible for this performance gap. We verified this hypothesis by introducing a set of heuristics which added more structure to the French parse trees. The parse trees modified by these heuristics proved to be able to significantly boost the performance of the QE system built upon them. However, their effect proved to be dependent on the data set used, as we observed only a small improvement with the SymForum data set.

In Chapter 6, we explored various ways to reach an optimum solution for semantic role labelling of French required, to compensate for the shortage of appropriate resources for French SRL. We first used a large artificially generated large corpus of French SRL annotation, provided via projection of English SRL over the Europarl parallel corpus through word alignments. We first tried to answer the following research question:

> *How much artificial data is needed to train an SRL system?*

The learning curves show that only a small fraction of this data was as effective as a much larger set in training a SRL system. We then set out to find a better projected annotation by answering the following research question:

*Is there a way to improve the projected annotation?*

We first used only direct translations of the Europarl corpus for projection, to reduce the adverse effect of translation shifts. However, the resulting projected annotations did not prove to be any more useful for training. Similarly, replacing the syntactic annotation of the projected SRL with universal POS tags and dependency labels did not significantly affect the results This, nevertheless, suggested that such annotations are interchangeable in the context of semantic role labelling. Moreover, we observed that the intersection of the word alignments of the two translation directions, which are commonly used in annotation projection to reduce noise, were too restrictive. We tried the union of these alignments as well as only source-to-target alignments leading to a significant increase in recall (and consequently f-score) both when used to create training data and when used in direct projection.

Despite these improvements, the resulting performance did not seem to be sufficient for the purpose of being used in semantic-based QE. We therefore put forth another research question:

*Is a large set of this artificial data better than a small set of hand-annotated data for training a SRL system?*

To answer this question, we used the available hand-annotated data set of 1K sentences to train an SRL system and compared it to the best-performing system trained on the projected annotations. The resulting system significantly outperformed the one built using the large set of synthetically generated data. We observed a same behaviour even when only a fraction of this small data was used for training, showing that hand-annotated data better suits this purpose, regardless of its quantity.

In Chapter 7, we investigated the use of semantic role labelling in translation quality estimation from various perspectives. We first addressed the following research question:

> *What is the most effective method of incorporating this semantic knowledge in QE?*

To find an answer for this question, we first used tree kernels to build a semantic-based QE system and examined various formats for encoding the semantic predicate-argument structure in trees to be used by this learning mechanism. The semantic information were most useful when used to augment the syntactic trees. We also designed a set of hand-crafted features extracted from semantic role labelling of source and target. Similar to syntax-based QE systems, the tree kernels outperformed the hand-crafted features. Moreover, we introduced PAM, a metric for estimating translation quality which used the predicate-argument structure match between the source and target, measured by different means including word alignments and translation tables, with word alignments proving to be more useful. Although the PAM scores showed a higher correlation with translation quality than the semantic hand-crafted features, we found them more useful when used as such features themselves. However, they did not appear to be a reliable estimator of translation quality. These observations shed light on the following research question:

> *To what extent does the semantic predicate-argument structure match between source and target represent the translation quality?*

With various methods of encoding semantic information in QE in hand, we combined them to answer the following research questions:

> *How effective is semantic role labelling, in general, in quality estimation of machine translation both in comparison and in combination with other surface-driven features as well as the syntactic information?*

The combination of the hand-crafted features and the tree kernels, as our fully semantic-based QE system, could outperform the WMT baseline in predicting HTER

and fluency scores, but not HBLEU and adequacy. Nor could the combination of the semantic-based and syntax-based QE systems outperform the baseline for these metrics. However, combination of the resulting system with the baseline features was successful. The best performing QE system built in this thesis was the combination of the baseline features with the semantic-based system which included the hand-crafted semantic features and semantically augmented syntactic tree kernels.

Focusing on the PAM metric as an estimation of translation quality, we performed a set of manual error analyses to answer the next research question:

*What are the factors hindering the performance of semantic-based QE?*

These analyses showed that the quality of semantic role labelling has to be improved, especially in the target side where the French SRL is employed. The word alignments, on the other hand, did not appear to be a major problem in the performance of PAM metric. To sum up, we believe that the low quality of semantic role labelling as well as the imbalance between the quality of source and target SRL (due to the significantly lower performance of the French SRL compared to English) hinders the performance of the semantic-based QE system.

In Chapter 8, we created two treebanks by annotating sentences taken from English and French Norton forum text with constituency structure. The aim of building these treebanks was to study the syntactic characteristics of Norton forum text. For this purpose, we developed an annotation strategy which marked user errors on the parse trees, which enabled us to analyse the user error rate in this text. Specifically we were able to find the answer to the following research question:

*How noisy is the user-generated content of the Norton forum text?*

We found that such errors occur in only a small fraction of the data and that the majority of them are capitalisation errors due to frequent product names and other proper nouns in the data. Missing punctuation appeared to be the second most frequent category of error. Once the level of user errors in the data was revealed, we asked the following research question:

*To what extent do user errors in the forum text affect its parse quality?*

To answer this question, we used the error marking to extract the correct form of the sentences. We then parsed the corrected sentences and found a modest improvement in the parsing performance. Despite its small size, relative to the extent of user error in the data, this improvement was noticeable showing that user errors do have a role to play in deterioration of parse quality.

The parsers used to parse the SymForum data in QE experiments were trained on edited newswire text, thus out-of-domain and out-of-style to Norton forum text. This raises the following research question:

*How noisy is out-of-domain parsing of the Norton forum text?*

To find the extent of noise in the parses of Norton forum text produced using parsers trained on newswire, we evaluated them on the Foreebanks. We observed a drastic performance drop on both English and French Foreebank. Based on the relatively small effect of user errors we found earlier on the parsing performance, it seems that the main contributing factors to the performance degradation when moving from newswire to the Norton forum text are problems such as unknown words and out-of-domain syntactic constructions such as questions and imperatives.

The low quality of Norton forum parsing is likely to affect the quality of its semantic role labelling which will in turn can hurt the performance of semantic-based QE as we observed in Chapter 7. The last research question we address in this thesis is concerned with this phenomenon:

*How effectively can we adapt our parsers to the Norton forum text, both intrinsically and in terms of the accuracy of semantic-based QE which uses semantic role labels from the new syntactic parse trees?*

To answer this question, we first used Foreebanks to adapt the parsers to the Norton forum text by providing them as supplementary training data on top of the original training data of the parsers. The augmented training data proved to be useful as all the adapted parsers outperformed the original ones. We then applied the

improved parsers to the SymForum data and semantic role labelled them using the new parses, to rebuild the semantic-based QE system and the PAM metric. While the accuracy of the metric and the system in predicting most of the evaluation metrics improved, only some of the changes were statistically significant. Considering the modest increase in parsing accuracy which led to these improvements, we can anticipate that using more sophisticated adaptation approaches will result in further improvements in semantic-based quality estimation.

## 9.2 Contributions

In this section, we briefly describe the contributions of this thesis as follows:

- We automatically compared the output of five different machine translation methods and found that these systems generate sufficiently different, though not systematically different, outputs for a sentence so that the combination of those systems is substantially better than each individual system.
- We also manually compared the output of a hierarchical phrase-based and a string-to-tree syntax-based SMT system and found that the syntax-based translation method used by this system showed no advantage in handling syntactic phenomena in translation such as long distance reordering.
- We created and publicly released *SymForum*, a data set for quality estimation of machine translated forum text, which contains 4,500 machine-translated sentence pairs post-edited and manually evaluated in terms of fluency and adequacy.
- We introduced a set of hand-crafted quality estimation features extracted from constituency and dependency parse trees.
- We built a syntax-based quality estimation system which could significantly outperform a well known baseline system when used with newswire data. This system combined successfully with that baseline system across the board when

used with the newswire data and for some quality metrics when used with forum content.

- We investigated different ways of building a semantic role labelling system for French and found that a very small set of manually annotated sentences was substantially more useful than a huge set of synthetically labelled sentences using a commonly used projection method.
- We also found that the intersection of the word alignments in the two translation directions, which is commonly used in annotation projection to reduce the noise level, is too restrictive and leads to a poorer projection quality than union or source-to-target alignments.
- We introduced a set of hand-crafted features extracted from the semantic role labelling of the source and target to be used in quality estimation. We also examined various ways of encoding the semantic role labelling in constituency and dependency trees to be used in tree kernel-based quality estimation.
- We introduced *PAM*, a new metric for quality estimation, which uses the predicate-argument structure match between a source sentence and its translation as a measure of translation quality. Our analysis showed that semantic role labelling errors, especially in the target, is the main factor negatively affecting the PAM accuracy.
- We built a semantic-based QE system which could outperform the baseline in predicting HTER and Fluency scores. The combination of this system with the baseline led to our best performing QE system.
- We built two treebanks named *Foreebank*, each containing 1000 sentences selected from English and French Norton forum text and manually annotated with their phrase structure syntax. We adopted an annotation strategy which accounts for the particularities of the forum text such as user errors and text styling.
- The Foreebank annotation strategy enabled us to analyse the type and level of user error in Norton forum text and its effect on parsing, finding that such

errors occur in only a small fraction of the text but correcting these few errors can noticeably improve the parsing performance.

- We successfully used the Foreebanks, despite their small sizes, to supplement the original training data of the English and French parsers as a way of adapting them to this text domain and type.
- We found that the improved parses of SymForum QE data using the adapted parses can improve the semantic-based QE systems and the PAM metric, but further improvement in SRL, especially on the French side to establish a balance with the English side, is required to observe a substantial improvement in QE.

## 9.3 Future Work

The work presented in this thesis could be extended in several directions. We used the syntax-based translation methods implemented in the Moses toolkit in Chapter 2. Other methods such as forest-based translation (Mi et al., 2008) or fuzzy use of syntax (Chiang, 2010; Zhang et al., 2011) or tools such as the *Joshua* decoder (Post et al., 2013) can also be experimented with in comparing syntax-based and phrase-based methods. It would also be interesting to examine the effect of parsing accuracy in the quality of syntax-based translation methods as we did for quality estimation as Neubig and Duh (2014) find that parser accuracy is important in their tree-to-sting translation setting. The availability of Foreebank can be a further help for such experiments. As to the combination of these two translation methods, one could investigate the use of quality estimation in selecting the best translation from the output of combined systems. Of course, the success of the combination will be strongly dependent on the reliability of the QE system.

In building the SymForum data set, we used three human evaluators to judge the translations which enabled us not only to reduce the degree of subjectivity and to increase reliability of the scores by averaging them, but also to compute

the agreement between the human evaluators on judging the translation quality. However, we only used one post-edit per translation to compute the human-targeted scores. It would be useful to know the extent of the post-editor agreement if more than one of them were employed for this purpose.

We combined the syntax- and semantic-based quality estimation systems built in this thesis with the baseline features of the WMT QE shared task. There has been considerable progress in extracting new non-syntactic features since the introduction of these features (Rubino et al., 2013). The complementarity of such new features with the syntax- and semantic-based QE systems of this thesis could also be examined. In addition applying these syntax- and semantic-based QE methods as well as the FTB modification heuristics to other data sets will shed additional lights on the generalizability of the conclusions made in this thesis.

Our semantic-based quality estimation was based on the semantic role labelling built on the dependency parses of the data. Therefore, only the translation of the head words of the syntactic constituents was taken into account in quality estimation. We used heuristics to transfer role labels to the constituent level, since constituency-based semantic role labelling is currently not possible due to the lack of resources for French. Therefore, using constituency-based semantic role labelling upon its availability is a reasonable next step for semantic-based QE. Furthermore, we were only able to use the verbal predicates for the same reason. Annotating the French SRL data set with nominal predicates will make it possible to use them in the semantic role labelling of both sides, which should in turn improve the performance of the QE systems. Finally, the semantic information used in this work was limited to the shallow predicate-argument representation which accounts for only part of the meaning of the sentence. The use of other types of semantic analysis in quality estimation, such as lexical distributional semantics, can also be investigated.

The main factor hindering the performance of semantic-based QE was shown to be the low quality of semantic role labelling, mainly on the target side, which also causes an imbalance between the two sides. The scarcity of French SRL resources is

the major obstacle to achieving a good quality labelling. Therefore, increasing the size of the available data set would appear to be a worthwhile effort.

We used a very simple method to adapt the parsers to Norton forum text. Using more sophisticated methods such as semi-supervised learning and system combination (Le Roux et al., 2012) is an avenue for further research. One aspect of parser adaptation is to improve the POS tagging as it has shown to be an important issue in parsing web text (Petrov and McDonald, 2012; Foster et al., 2011a). In addition, the annotation strategy of Foreebank marks the user errors on the parse trees. These annotations can also be useful in correcting user errors in this type of text.

# Bibliography


Abeillé, A., Clément, L., and Toussenel, F. (2003). Building a Treebank for French. In *Treebanks: Building and Using Syntactically Annotated Corpora*, pages 165–187. Kluwer Academic Publishers.

Al-Qinai, J. (2000). Translation quality assessment. strategies, parameters and procedures. *Meta*, XLV(3):497–519.

Allen, J. (2003). Post-editing. In Somers, H., editor, *Computers and Translation: A Translator's Guide*, pages 297–317. John Benjamins Publishing Company.

Alumäe, T. and Kurimo, M. (2010). Domain adaptation of maximum entropy language models. In *Proceedings of ACL: Short Papers*, pages 301–306.

Attia, M., Foster, J., Hogan, D., Roux, J. L., Tounsi, L., and van Genabith, J. (2010). Handling Unknown Words in Statistical Latent-Variable Parsing Models for Arabic, English and French. In *Proceedings of the 1st Workshop on SPMRL*, pages 67–75.

Auli, M., Lopez, A., Hoang, H., and Koehn, P. (2009). A Systematic Analysis of Translation Model Search Spaces. In *Proceedings of WMT*, pages 224–232.

Avramidis, E. (2012). Quality estimation for Machine Translation output using linguistic analysis and decoding features. In *Proceedings of WMT*, pages 84–90.

Baker, C. F., Fillmore, C. J., and Lowe, J. B. (1998). The Berkeley FrameNet Project. In *Proceedings of ACL*, pages 86–90.

Banerjee, P. (2013). *Domain Adaptation for Statistical Machine Translation of Corporate and User-Generated Content*. PhD thesis, Dublin City University.

Banerjee, P., Naskar, S. K., Roturier, J., Way, A., and van Genabith, J. (2012). Domain Adaptation in SMT of User-Generated Forum Content Guided by OOV Word Reduction: Normalization and/or Supplementary Data? In *Proceedings of EAMT*.

Bicici, E. and Way, A. (2014). Referential Translation Machines for Predicting Translation Quality. In *Proceedings of WMT*, pages 313–321.

Bies, A., Ferguson, M., Katz, K., MacIntyre, R., Tredinnick, V., Kim, G., Marcinkiewicz, M. A., and Schasberger, B. (1995). Bracketing Guidelines for Treebank II Style Penn Treebank Project. Technical report, University of Pennsylvania.

Björkelund, A., Hafdell, L., and Nugues, P. (2009). Multilingual Semantic Role Labeling. In *Proceedings of CoNLL: Shared Task*, pages 43–48.

Blatz, J., Fitzgerald, E., Foster, G., Gandrabur, S., Goutte, C., Kulesza, A., Sanchis, A., and Ueffing, N. (2004). Confidence Estimation for Machine Translation. In *Proceedings of COLING*.

Blitzer, J., McDonald, R., and Pereira, F. (2006). Domain Adaptation with Structural Correspondence Learning. In *Proceedings of EMNLP*, pages 120–128.

Bod, R. (2007). Is the End of Supervised Parsing in Sight? In *Proceedings of ACL*, pages 400–407.

Bojar, O., Buck, C., Callison-Burch, C., Federmann, C., Haddow, B., Koehn, P., Monz, C., Post, M., Soricut, R., and Specia, L. (2013). Findings of the 2013 workshop on statistical machine translation. In *Proceedings of WMT*, pages 1–44.

Bojar, O., Buck, C., Federmann, C., Haddow, B., Koehn, P., Leveling, J., Monz, C., Pecina, P., Post, M., Saint-Amand, H., Soricut, R., Specia, L., and Tamchyna, A. (2014). Findings of the 2014 Workshop on Statistical Machine Translation. In *Proceedings of WMT*, pages 12–58.

Bojar, O. and Wu, D. (2012). Towards a Predicate-argument Evaluation for MT. In *Proceedings of SSST-6*, pages 30–38.

Brants, S., Dipper, S., Hansen, S., Lezius, W., and Smith, G. (2002). The TIGER Treebank. In *Proceedings of the Workshop on Treebanks and Linguistic Theories*.

Brown, P., Cocke, J., Pietra, S. D., Pietra, V. D., Jelinek, F., Mercer, R., and Roossin, P. (1988). A Statistical Approach to Language Translation. In *Proceedings of the 12th Conference on Computational Linguistics - Volume 1*, pages 71–76.

Butt, M., Dyvik, H., King, T. H., Masuichi, H., and Rohrer, C. (2002). The parallel grammar project. In *Proceedings of the 2002 workshop on Grammar engineering and evaluation-Volume 15*, pages 1–7.

Callison-Burch, C., Fordyce, C., Koehn, P., Monz, C., and Schroeder, J. (2007). (Meta-) Evaluation of Machine Translation. In *Proceedings of WMT*, pages 136–158.

Callison-Burch, C., Koehn, P., Monz, C., Post, M., Soricut, R., and Specia, L. (2012). Findings of the 2012 Workshop on Statistical Machine Translation. In *Proceedings of WMT*, pages 10–51.

Candito, M., Crabbé, B., and Denis, P. (2010). Statistical French dependency parsing: treebank conversion and first results. In *Proceedings of LREC'2010*, pages 1840–1847.

Candito, M. and Crabbé, B. (2009). Improving Generative Statistical Parsing with Semi-supervised Word Clustering. In *Proceedings of IWPT*, pages 138–141.

Carreras, X. and Collins, M. (2009). Non-Projective Parsing for Statistical Machine Translation. In *Proceedings of EMNLP*, pages 200–209.

Charniak, E. (2001). Immediate-head Parsing for Language Models. In *Proceedings of ACL*, pages 124–131.

Charniak, E. and Johnson, M. (2005). Course-to-fine n-best-parsing and MaxEnt discriminative reranking. In *Proceedings of ACL*, pages 173–180.

Chiang, D. (2007). Hierarchical Phrase-based Translation. *Computational Linguistics*, 33(2):201–228.

Chiang, D. (2010). Learning to Translate with Source and Target Syntax. In *Proceedings of ACL*, pages 1443–1452.

Collins, M. (1997). Three Generative, Lexicalised Models for Statistical Parsing. In *Proceedings of ACL-EACL*, pages 16–23.

Collins, M. (1999). *Head-driven Statistical Models for Natural Language Parsing*. PhD thesis, University of Pennsylvania.

Collins, M. and Duffy, N. (2002). New Ranking Algorithms for Parsing and Tagging: Kernels over Discrete Structures, and the Voted Perceptron. In *Proceedings of ACL*, pages 263–270.

Daumé, III, H., Kumar, A., and Saha, A. (2010). Frustratingly Easy Semi-supervised Domain Adaptation. In *Proceedings of the 2010 Workshop on Domain Adaptation for Natural Language Processing*, pages 53–59.

Daumé III, H. and Marcu, D. (2006). Domain Adaptation for Statistical Classifiers. *Journal of Artificial Intelligence Research (JAIR)*, 26:101–126.

De Almeida, G. and O'Brien, S. (2010). Analysing post-editing performance: correlations with years of translation experience. In *Proceedings of EAMT*.

de Marneffe, M.-C. and Manning, C. D. (2008). The Stanford typed dependencies representation. In *Proceedings of the COLING Workshop on Cross-Framework and Cross-Domain Parser Evaluation*, pages 1–8.

DeNeefe, S., Knight, K., Wang, W., and Marcu, D. (2007). What can syntax-based MT learn from phrase-based MT? In *Proceedings of EMNLP-CoNLL*, pages 755–763.

Denis, P. and Sagot, B. (2012). Coupling an Annotated Corpus and a Lexicon for State-of-the-art POS Tagging. *Language Resources and Evaluation*, 46(4):721–736.

Denkowski, M. and Lavie, A. (2011). Meteor 1.3: Automatic Metric for Reliable Optimization and Evaluation of Machine Translation Systems. In *Proceedings of WMT*, pages 85–91.

Dugast, L., Senellart, J., and Koehn, P. (2007). Statistical post-editing on SYSTRAN's rule-based translation system. In *Proceedings of WMT*, pages 220–223.

Efron, B. and Tibshirani, R. J. (1993). *An Introduction to the Bootstrap*. Chapman & Hall.

Finkel, J. R. and Manning, C. D. (2009). Hierarchical bayesian domain adaptation. In *Proceedings of NAACL*, pages 602–610.

Foster, J. (2010). cba to check the spelling investigating parser performance on discussion forum posts. In *Proceedings of NAACL*, pages 381–384.

Foster, J., Çetinoğlu, Ö., Wagner, J., Roux, J. L., Nivre, J., Hogan, D., and van Genabith, J. (2011a). From News to Comment: Benchmarks and Resources for Parsing the Language of Web 2.0. In *Proceedings of IJCNLP*, pages 893–901.

Foster, J., Çetinoğlu, Ö., Wagner, J., and van Genabith, J. (2011b). Comparing the use of edited and unedited text in parser self-training. In *Proceedings of IWPT*, pages 215–219.

Foster, J. and van Genabith, J. (2008). Parser Evaluation and the BNC: Evaluating 4 constituency parsers with 3 metrics. In *Proceedings LREC*.

Foster, J., Wagner, J., and Van Genabith, J. (2008). Adapting a WSJ-trained parser to grammatically noisy text. In *Proceedings of ACL: Short Papers*, pages 221–224.

Fox, H. J. (2002). Phrasal cohesion and statistical machine translation. In *Proceedings of EMNLP*, pages 304–311.

Galley, M., Hopkins, M., Knight, K., and Marcu, D. (2004). What's in a translation rule? In *Proceedings of HLT-NAACL*.

Gamon, M., Aue, A., and Smets, M. (2005). Sentence-Level MT evaluation without reference translations: beyond language modeling. In *Proceedings of EAMT*, pages 103–111.

Gandrabur, S. and Foster, G. (2003). Confidence estimation for translation prediction. In *Proceedings of CoNLL*, pages 95–102.

Gardent, C. and Cerisara, C. (2010). Semi-Automatic Propbanking for French. In *TLT9 - The Ninth International Workshop on Treebanks and Linguistic Theories*, pages 67–78.

Gildea, D. (2001). Corpus Variation and Parser Performance. In *Proceedings of EMNLP*, pages 167–202.

Gildea, D. and Jurafsky, D. (2002). Automatic Labelling of Semantic Roles. *Computational Linguistics*, 28(3):245–288.

Giménez, J. and Màrquez, L. (2007). Linguistic Features for Automatic Evaluation of Heterogenous MT Systems. In *Proceedings of WMT*, pages 256–264.

Giménez, J. and Màrquez, L. (2010). Asiya: An Open Toolkit for Automatic Machine Translation (Meta-)Evaluation. *Prague Bull. Math. Linguistics*, 94:77–86.

Goto, I., Utiyama, M., Onishi, T., and Sumita, E. (2011). A Comparison Study of Parsers for Patent Machine Translation. In *Proceedings of MT Summit*, pages 46–54.

Hajič, J., Ciaramita, M., Johansson, R., Kawahara, D., Martí, M. A., Màrquez, L., Meyers, A., Nivre, J., Padó, S., Štěpánek, J., Straňák, P., Surdeanu, M., Xue, N., and Zhang, Y. (2009). The CoNLL-2009 Shared Task: Syntactic and Semantic Dependencies in Multiple Languages. In *Proceedings of CoNLL: Shared Task*, pages 1–18.

Hamon, O., Hartley, A., Popescu-Belis, A., and Choukri, K. (2007). Assessing human and automated quality judgments in the French MT evaluation campaign CESTA. In *Proceedings of the MT Summit XI*, pages 231–238.

Hardmeier, C., Nivre, J., and Tiedemann, J. (2012). Tree Kernels for Machine Translation Quality Estimation. In *Proceedings of WMT*, pages 109–113.

Heafield, K., Koehn, P., and Lavie, A. (2013). Grouping Language Model Boundary Words to Speed K–Best Extraction from Hypergraphs. In *Proceedings of NAACL-HLT*, pages 958–968.

Heidorn, G. (2000). Intelligent writing assistance. *Handbook of Natural Language Processing*, pages 181–207.

Hemphill, C. T., Godfrey, J. J., and Doddington, G. R. (1990). The ATIS Spoken Language Systems Pilot Corpus. In *Proceedings of the Workshop on Speech and Natural Language*, pages 96–101.

Hoang, H., Koehn, P., and Lopez, A. (2009). A unified framework for phrase-based, hierarchical, and syntax-based statistical machine translation. In *Proceedings of IWSLT*, pages 152–159.

Huang, F. and Papineni, K. (2007). Hierarchical System Combination for Machine Translation. In *Proceedings of EMNLP-CoNLL*, pages 277–286.

Huang, L. and Chiang, D. (2007). Forest Rescoring: Faster Decoding with Integrated Language Models. In *Proceedings of ACL*, pages 144–151.

Huang, L., Knight, K., and Joshi, A. (2006). Statistical syntax-directed translation with extended domain of locality. In *Proceedings of AMTA*, pages 66–73.

Huang, Z. and Harper, M. (2009). Self-training PCFG Grammars with Latent Annotations Across Languages. In *Proceedings of EMNLP*.

Hwa, R., Resnik, P., Weinberg, A., Cabezas, C., and Kolak, O. (2005). Bootstrapping parsers via syntactic projection across parallel texts. *Natural language engineering*, 11(03):311–325.

Johansson, R. and Nugues, P. (2007). Extended Constituent-to-dependency conversion for English. In *Proceedings of NODALIDA*, pages 105–112.

Judge, J., Cahill, A., and Van Genabith, J. (2006). Questionbank: Creating a corpus of parse-annotated questions. In *Proceedings of COLING-ACL*, pages 497–504.

Kaljahi, R. S. Z., Foster, J., and Roturier, J. (2014a). Semantic Role Labelling with minimal resources: Experiments with French. In *Proceedings of *SEM*, pages 87–92.

Kaljahi, R. S. Z., Foster, J., and Roturier, J. (2014b). Syntax and Semantics in Quality Estimation of Machine Translation. In *Proceedings of SSST-8*, pages 67–77.

Kaljahi, R. S. Z., Foster, J., Rubino, R., and Roturier, J. (2014c). Quality Estimation of English-French Machine Translation: A Detailed Study of the Role of Syntax. In *Proceedings of COLING*, pages 2052–2063.

Kaljahi, R. S. Z., Foster, J., Rubino, R., Roturier, J., and Hollowood, F. (2013). Parser Accuracy in Quality Estimation of Machine Translation: A Tree Kernel Approach. In *Proceedings of IJCNLP*, pages 1092–1096.

Kaljahi, R. S. Z., Rubino, R., Roturier, J., and Foster, J. (2012). A detailed analysis of phrase-based and syntax-based machine translation: the search for systematic differences. In *Proceedings of AMTA*.

Klein, D. and Manning, C. D. (2003). Accurate Unlexicalized Parsing. In *Proceedings of ACL*, pages 423–430.

Koehn, P. (2005). Europarl: A Parallel Corpus for Statistical Machine Translation. In *Proceedings of MT Summit*, pages 79–86.

Koehn, P., Och, F. J., and Marcu, D. (2003). Statistical Phrase-based Translation. In *Proceedings of NAACL-HLT*, pages 48–54.

Kuĉera, H. and Francis, W. N. (1967). *Computational Analysis of Present-Day American English*. Brown University Press.

Kupść, A. and Abeillé, A. (2008). Growing TreeLex. In *Proceedings of CICLing*, pages 28–39.

LDC (2002). Linguistic Data Annotation Specification: Assessment of Fluency and Adequacy in Chinese-English Translations. Technical report.

Le Roux, J., Foster, J., Wagner, J., Kaljahi, R. S. Z., and Bryl, A. (2012). DCU-Paris13 Systems for the SANCL 2012 Shared Task. In *Working Notes of SANCL*, pages 1–4.

Li, X. and Roth, D. (2002). Learning question classifiers. In *Proceedings of COLING*, pages 1–7.

Liu, D. and Gildea, D. (2005). Syntactic Features for Evaluation of Machine Translation. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 25–32.

Liu, D. and Gildea, D. (2010). Semantic role features for machine translation. In *Proceedings of COLING*, pages 716–724.

Lo, C.-k., Beloucif, M., Saers, M., and Wu, D. (2014). XMEANT: Better semantic MT evaluation without reference translations. In *Proceedings of ACL: Short Papers*, pages 765–771.

Lo, C.-k., Tumuluru, A. K., and Wu, D. (2012). Fully Automatic Semantic MT Evaluation. In *Proceedings of WMT*, pages 243–252.

Lo, C.-k. and Wu, D. (2011). MEANT: An Inexpensive, High-accuracy, Semi-automatic Metric for Evaluating Translation Utility via Semantic Frames. In *Proceedings of ACL*, pages 220–229.

Lo, C.-k. and Wu, D. (2013). Meant at wmt 2013: A tunable, accurate yet inexpensive semantic frame based mt evaluation metric. In *Proceedings of WMT*, pages 422–428.

Lorenzo, A. and Cerisara, C. (2012). Unsupervised frame based Semantic Role Induction: application to French and English. In *Proceedings of the ACL 2012 Joint Workshop on Statistical Parsing and Semantic Processing of Morphologically Rich Languages*, pages 30–35.

Magerman, D. M. (1995). Statistical Decision-tree Models for Parsing. In *Proceedings of ACL*, pages 276–283.

Marcu, D., Wang, W., Echihabi, A., and Knight, K. (2006). SPMT: Statistical Machine Translation with Syntactified Target Language Phrases. In *Proceedings of EMNLP*, pages 44–52.

Marcus, M., Kim, G., Marcinkiewicz, M. A., MacIntyre, R., Bies, A., Ferguson, M., Katz, K., and Schasberger, B. (1994). The Penn Treebank: Annotating Predicate Argument Structure. In *Proceedings of the 1994 ARPA Speech and Natural Language Workshop*, pages 114–119.

Marcus, M. P., Santorini, B., and Marcinkiewicz, M. A. (1993). Building a Large Annotated Corpus of English: the Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Màrquez, L., Carreras, X., Litkowski, K. C., and Stevenson, S. (2008). Semantic Role Labeling: An Introduction to the Special Issue. *Computational Linguistics*, 34(2):145–159.

Maxwell, J. and Kaplan, R. (1996). Unification-based parsers that automatically take advantage of context freeness. In *LFG96 Conference, Grenoble, France. Ms. Xerox PARC*.

McClosky, D. (2010). *Any Domain Parsing: Automatic Domain Adaptation for Natural Language Parsing*. PhD thesis, Brown University.

McClosky, D., Charniak, E., and Johnson, M. (2006). Effective Self-training for Parsing. In *Proceedings of HLT-NAACL*, pages 152–159.

McClosky, D., Charniak, E., and Johnson, M. (2010). Automatic Domain Adaptation for Parsing. In *Proceedings of HLT-NAACL*, pages 28–36.

McDonald, R., Lerman, K., and Pereira, F. (2006). Multilingual dependency analysis with a two-stage discriminative parser. In *Proceedings of CoNLL*, pages 216–220.

McDonald, R., Nivre, J., Quirmbach-Brundage, Y., Goldberg, Y., Das, D., Ganchev, K., Hall, K., Petrov, S., Zhang, H., Täckström, O., Bedini, C., Bertomeu Castelló, N., and Lee, J. (2013). Universal Dependency Annotation for Multilingual Parsing. In *Proceedings of ACL: Short Papers*, pages 92–97.

Mi, H., Huang, L., and Liu, Q. (2008). Forest-Based Translation. In *Proceedings of ACL-08: HLT*, pages 192–199.

Miyao, Y., Saetre, R., Sagae, K., Matsuzaki, T., and Tsujii, J. (2008). Task-oriented Evaluation of Syntactic Parsers and their Representations. In *Proceedings of ACL*, pages 46–54.

Mollá, D. and Hutchinson, B. (2003). Intrinsic versus Extrinsic Evaluation of Parsing Systems. In *Proceedings of EACL*, pages 43–50.

Moschitti, A. (2006). Making Tree Kernels practical for Natural Language Learning. In *Proceedings of EACL*, pages 113–120.

Moschitti, A., Pighin, D., and Basili, R. (2006). Tree kernel engineering for proposition re-ranking. In *Proceedings of Mining and Learning with Graphs (MLG)*, pages 165–172.

Mott, J., Bies, A., Laury, J., and Warner, C. (2012). Bracketing Webtext: An Addendum to Penn Treebank II Guidelines. Technical report, Linguistic Data Consortium.

Mott, J., Bies, A., Warner, C., and Taylor, A. (2009). Supplementary Guidelines for ETTB 2.0. Technical report, Linguistic Data Consortium.

Neubig, G. and Duh, K. (2014). On the Elements of an Accurate Tree-to-String Machine Translation System. In *Proceedings of ACL*, pages 143–149.

Nivre, J., Hall, J., and Nilsson, J. (2006). MaltParser: A Data-Driven Parser-Generator for Dependency Parsing. In *Proceedings of LREC*, pages 2216–2219.

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

Och, F. and Ney, H. (2004). The alignment template approach to statistical machine translation. *Computational Linguistics*, 30(4):417–449.

Och, F. J. and Ney, H. (2003). A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, 29(1):19–51.

Och, F. J., Tillmann, C., Ney, H., and Informatik, L. F. (1999). Improved Alignment Models for Statistical Machine Translation. In *Proceedings of the Joint EMNLP and Very Large Corpora*, pages 20–28.

Owczarzak, K. (2008). *A novel dependency-based evaluation metric for Machine Translation*. PhD thesis, Dublin City University.

Padó, S. and Lapata, M. (2009). Cross-lingual Annotation Projection of Semantic Roles. *Journal of Artificial Intelligence Research*, 36(1):307–340.

Palmer, M., Gildea, D., and Kingsbury, P. (2005). The Proposition Bank: An Annotated Corpus of Semantic Roles. *Computational Linguistics*, 31(1):71–106.

Papineni, K., Roukos, S., Ward, T., and Zhu, W. (2002). BLEU: a method for automatic evaluation of machine translation. In *Proceedings of ACL*, pages 311–318.

Petrov, S. (2009). *Coarse-to-fine Natural Language Processing*. PhD thesis, University of California, Berkeley.

Petrov, S., Barrett, L., Thibaux, R., and Klein, D. (2006). Learning Accurate, Compact and Interpretable Tree Annotation. In *Proceedings of COLING-ACL*.

Petrov, S., Das, D., and McDonald, R. (2012). A Universal Part-of-Speech Tagset. In *Proceedings of LREC*.

Petrov, S. and McDonald, R. (2012). Overview of the 2012 shared task on parsing the web. *Notes of the First Workshop on Syntactic Analysis of Non-Canonical Language (SANCL)*, 59.

Pierce, J. R. and Carroll, J. B. (1966). *Language and Machines: Computers in Translation and Linguistics*. National Academy of Sciences/National Research Council, Washington, DC, USA.

Pighin, D. and Màrquez, L. (2011). Automatic Projection of Semantic Structures: An Application to Pairwise Translation Ranking. In *Proceedings of SSST*, pages 1–9.

Plank, B. (2011). *Domain Adaptation for Parsing*. Ph.d. thesis, University of Groningen.

Post, M., Ganitkevitch, J., Orland, L., Weese, J., Cao, Y., and Callison-Burch, C. (2013). Joshua 5.0: Sparser, Better, Faster, Server. In *Proceedings of WMT*, pages 206–212.

Punyakanok, V., Roth, D., and Yih, W.-t. (2008). The importance of syntactic parsing and inference in semantic role labeling. *Computational Linguistics*, 34(2):257–287.

Quirk, C. (2004). Training a Sentence-Level Machine Translation Confidence Measure. In *Proceedings of LREC*, pages 825–828.

Quirk, C. and Corston-Oliver, S. (2006). The impact of parse quality on syntactically-informed statistical machine translation. In *Proceedings of EMNLP*, pages 62–69.

Ravi, S., Knight, K., and Soricut, R. (2008). Automatic Prediction of Parser Accuracy. In *Proceedings of EMNLP*, pages 887–896.

Roturier, J. and Bensadoun, A. (2011). Evaluation of MT Systems to Translate User Generated Content. pages 244–251.

Rubino, R., Foster, J., Kaljahi, R. S. Z., Roturier, J., and Hollowood, F. (2013). Estimating the quality of translated user-generated content. In *Proceedings of IJCNLP*, pages 1167–1173.

Rubino, R., Foster, J., Wagner, J., Roturier, J., Kaljahi, R., and Hollowood, F. (2012). DCU-Symantec Submission for the WMT 2012 Quality Estimation Task. In *Proceedings of WMT*, pages 138–144.

Schluter, N. and van Genabith, J. (2007). Preparing, Restructuring, and Augmenting a French Treebank: Lexicalised Parsers or Coherent Treebanks? In *Proceedings of the 10th Conference of the Pacific Association for Computational Linguistics*.

Schuler, K. K. (2006). *VerbNet: A Broad-Coverage, Comprehensive Verb Lexicon*. PhD thesis, University of Pennsylvania.

Seddah, D., Sagot, B., Candito, M., Mouilleron, V., and Combet, V. (2012). The French Social Media Bank: a Treebank of Noisy User Generated Content. In *Proceedings of COLING*, pages 2441–2458.

Shen, L., Xu, J., and Weischedel, R. (2010). String-to-dependency Statistical Machine Translation. *Computational Linguistics*, 36(4):649–671.

Snover, M., Dorr, B., Schwartz, R., Micciulla, L., and Makhoul, J. (2006). A study of translation edit rate with targeted human annotation. In *Proceedings of AMTA*.

Soricut, R., Bach, N., and Wang, Z. (2012). The SDL Language Weaver Systems in the WMT12 Quality Estimation Shared Task. In *Proceedings of WMT*, pages 145–151.

Soricut, R. and Echihabi, A. (2010). TrustRank: Inducing Trust in Automatic Translations via Ranking. In *Proceedings of ACL*, pages 612–621.

Specia, L. and Giménez, J. (2010). Combining confidence estimation and reference-based metrics for segment level mt evaluation. In *Proceedings of AMTA*.

Specia, L., Shah, K., de Souza, J. G., and Cohn, T. (2013). QuEst - A translation quality estimation framework. In *ACL: System Demonstrations*, pages 79–84.

Specia, L., Turchi, M., Wang, Z., Shawe-Taylor, J., and Saunders, C. (2009). Improving the confidence of machine translation quality estimates. In *Proceedings of MT Summit XII*, pages 73–80.

Surdeanu, M., Johansson, R., Meyers, A., Màrquez, L., and Nivre, J. (2008). The CoNLL-2008 Shared Task on Joint Parsing of Syntactic and Semantic Dependencies. In *Proceedings of CoNLL*, pages 159–177.

Tateisi, Y., Yakushiji, A., and Ohta, T. (2005). Syntax annotation for the GENIA corpus. In *Proceedings of IJCNLP*, pages 220–225.

Taylor, A. (1996). Bracketing Switchboard: An Addendum to the Treebank II Guidelines. Technical report.

Titov, I. and Henderson, J. (2007). A Latent Variable Model for Generative Dependency Parsing. In *Proceedings of IWPT*, pages 144–155.

Titov, I., Henderson, J., Merlo, P., and Musillo, G. (2009). Online Projectivisation for Synchronous Parsing of Semantic and Syntactic Dependencies. In *Proceedings of IJCAI*, pages 1562–1567.

Tu, Z., He, Y., Foster, J., van Genabith, J., Liu, Q., and Lin, S. (2012). Identifying High-Impact Sub-Structures for Convolution Kernels in Document-level Sentiment Classification. In *Proceedings of ACL*, pages 338–343.

Turian, J., Shen, L., and Melamed, I. D. (2003). Evaluation of Machine Translation and its Evaluation. In *Proceedings of MT Summit IX*, pages 386–393.

Ueffing, N., Macherey, K., and Ney, H. (2003). Confidence measures for statistical machine translation. In *Proceedings of MT Summit IX*.

Ueffing, N., Och, F. J., and Ney, H. (2002). Generation of Word Graphs in Statistical Machine Translation. In *Proceedings of EMNLP*, pages 156–163.

van der Plas, L., Henderson, J., and Merlo, P. (2010a). D6. 2: Semantic Role Annotation of a French-English Corpus.

van der Plas, L., Merlo, P., and Henderson, J. (2011). Scaling up Automatic Cross-Lingual Semantic Role Annotation. In *Proceedings of ACL-HLT*, pages 299–304.

van der Plas, L., Samardžić, T., and Merlo, P. (2010b). Cross-lingual Validity of PropBank in the Manual Annotation of French. In *Proceedings of the Fourth Linguistic Annotation Workshop*, LAW IV '10, pages 113–117.

Wagner, J., Foster, J., and van Genabith, J. (2009). Judging grammaticality: Experiments in sentence classification. *CALICO Journal*, 26(3):474–490.

Warner, C., Bies, A., Brisson, C., and Mott, J. (2004). Addendum to the Penn Treebank II Style bracketing Guidelines: BioMedical Treebank Annotation. Technical report, University of Pennsylvania.

Wiegand, M. and Klakow, D. (2010). Convolution Kernels for Opinion Holder Extraction. In *Proceedings of NAACL-HLT*, pages 795–803.

Wu, D. and Fung, P. (2009). Can semantic role labeling improve SMT. In *Proceedings of EAMT*, pages 218–225.

Wu, D. and Wong, H. (1998). Machine translation with a stochastic grammatical channel. In *Proceedings of ACL*, pages 1408–1414.

Xiong, D., Liu, Q., and Lin, S. (2007). A dependency treelet string correspondence model for statistical machine translation. In *Proceedings of WMT*, pages 40–47.

Yamada, K. and Knight, K. (2001). A syntax-based statistical translation model. In *Proceedings of ACL*, pages 523–530.

Yarowsky, D. (1995). Unsupervised Word Sense Disambiguation Rivaling Supervised Methods. In *Proceedings of ACL*, pages 189–196.

Yu, H., Wu, X., Xie, J., Jiang, W., Liu, Q., and Lin, S. (2014). RED: A Reference Dependency Based MT Evaluation Metric. In *Proceedings of COLING*, pages 2042–2051.

Zhang, H., Wang, H., Xiao, T., and Zhu, J. (2010). The impact of parsing accuracy on syntax-based SMT. In *Proceedings of the International Conference on NLP-KE*.

Zhang, J., Zhai, F., and Zong, C. (2011). Augmenting string-to-tree translation models with fuzzy use of source-side syntax. In *Proceedings of EMNLP*, pages 204–215.

Zollmann, A. and Venugopal, A. (2006). Syntax Augmented Machine Translation via Chart Parsing. In *Proceedings of WMT*, pages 138–141.

Zollmann, A., Venugopal, A., Och, F., and Ponte, J. (2008). A Systematic Comparison of Phrase-Based, Hierarchical and Syntax-Augmented Statistical MT. In *Proceedings of COLING*, pages 1145–1152.

Zwarts, S. and Dras, M. (2008). Choosing the Right Translation: A Syntactically Informed Classification Approach. In *Proceedings of COLING*, pages 1153–1160.

# Appendix A

# Quality Estimation Results

Table A.1: Quality estimation results using the SymForum data set

| | 1-HTER | | HBLEU | | Adequacy | | Fluency | |
|---|---|---|---|---|---|---|---|---|
| | RMSE | r | RMSE | r | RMSE | r | RMSE | r |
| | Baseline QE systems | | | | | | | |
| B-Mean | 0.2442 | - | 0.2907 | - | 0.2501 | - | 0.2796 | - |
| B-WMT17 | 0.2310 | 0.3661 | 0.2696 | 0.3806 | 0.2219 | 0.4710 | 0.2469 | 0.4769 |
| | Syntax-based QE systems | | | | | | | |
| SyTK | 0.2267 | 0.3693 | 0.2721 | 0.3559 | 0.2258 | 0.4306 | 0.2431 | 0.5013 |
| B+SyTK | 0.2243 | 0.3935 | 0.2655 | 0.4082 | 0.2215 | 0.4632 | 0.2403 | 0.5144 |
| SyHC | 0.2435 | 0.2572 | 0.2797 | 0.3080 | 0.2334 | 0.3961 | 0.2479 | 0.4696 |
| B+SyHC | 0.2265 | 0.4159 | 0.2689 | 0.4080 | 0.2221 | 0.4795 | 0.2387 | 0.5269 |
| SyQE | 0.2255 | 0.3824 | 0.2711 | 0.3650 | 0.2248 | 0.4393 | 0.2419 | 0.5087 |
| B+SyQE | 0.2236 | 0.4017 | 0.2686 | 0.3852 | 0.2219 | 0.4632 | 0.2391 | 0.5255 |
| SyQE$_L$ | 0.2273 | 0.3647 | 0.2731 | 0.3455 | 0.2249 | 0.4386 | 0.2415 | 0.5097 |
| SyTK/C-T$_m$ | 0.2302 | 0.3331 | 0.2778 | 0.3114 | 0.2265 | 0.4236 | 0.2444 | 0.4912 |
| SyTK/CD-ST$_m$ | 0.2257 | 0.3800 | 0.2715 | 0.3622 | 0.2253 | 0.4359 | 0.2425 | 0.5056 |
| SyTK/C-ST | 0.2292 | 0.3446 | 0.2749 | 0.3349 | 0.2266 | 0.4241 | 0.2442 | 0.4939 |
| | Semantic-based QE systems | | | | | | | |
| SeTK/D-PAS$_{POS}$ | 0.2489 | 0.1774 | 0.2856 | 0.1843 | 0.2423 | 0.2770 | 0.2652 | 0.3252 |
| SeTK/D-PAS$_{word}$ | 0.2480 | 0.1789 | 0.2926 | 0.1669 | 0.2485 | 0.2660 | 0.2654 | 0.3221 |
| SeTK/D-PST$_w$ | 0.2413 | 0.2082 | 0.2832 | 0.2237 | 0.2431 | 0.2567 | 0.2663 | 0.3080 |
| SeTK/D-PST$_{wdp}$ | 0.2409 | 0.2136 | 0.2815 | 0.2450 | 0.2383 | 0.3169 | 0.2606 | 0.3670 |
| SeTK/D-SAS$_{afx}$ | 0.2270 | 0.3699 | 0.2738 | 0.3391 | 0.2291 | 0.4022 | 0.2476 | 0.4731 |
| SeTK/D-SAS$_{node}$ | 0.2271 | 0.3667 | 0.2727 | 0.3483 | 0.2275 | 0.4169 | 0.2443 | 0.4930 |
| SyTK/D-ST | 0.2261 | 0.3778 | 0.2722 | 0.3546 | 0.2280 | 0.4118 | 0.2455 | 0.4860 |
| SeTK/C-PST | 0.2400 | 0.2319 | 0.2809 | 0.2541 | 0.2410 | 0.2966 | 0.2615 | 0.3616 |
| SeTK/C-SAS | 0.2289 | 0.3462 | 0.2744 | 0.3359 | 0.2261 | 0.4277 | 0.2441 | 0.4940 |
| SeTK/CD-PST | 0.2394 | 0.2311 | 0.2795 | 0.2714 | 0.2373 | 0.3303 | 0.2578 | 0.3923 |
| SSTK | 0.2269 | 0.3682 | 0.2722 | 0.3537 | 0.2253 | 0.4351 | 0.2425 | 0.5046 |
| B+SSTK | 0.2227 | 0.4104 | 0.2671 | 0.3948 | 0.2174 | 0.4957 | 0.2381 | 0.5273 |
| SeHC | 0.2482 | 0.1794 | 0.2868 | 0.1636 | 0.2416 | 0.2972 | 0.2612 | 0.3577 |
| SSHC | 0.2362 | 0.3107 | 0.2787 | 0.3027 | 0.2326 | 0.4009 | 0.2471 | 0.4726 |
| B+SeHC | 0.2310 | 0.3660 | 0.2697 | 0.3792 | 0.2275 | 0.4444 | 0.2439 | 0.4873 |
| B+SSHC | 0.2271 | 0.4066 | 0.2677 | 0.4030 | 0.2252 | 0.4658 | 0.2393 | 0.5234 |
| WAPAM-UPrec | 0.3325 | 0.1862 | 0.3851 | 0.1721 | 0.3319 | 0.2215 | 0.4334 | 0.2363 |
| WAPAM-URec | 0.3249 | 0.2483 | 0.3706 | 0.2338 | 0.3213 | 0.2796 | 0.4171 | 0.2927 |
| WAPAM-UF$_1$ | 0.3175 | 0.2328 | 0.3607 | 0.2179 | 0.3108 | 0.2698 | 0.4033 | 0.2865 |
| WAPAM-LPrec | 0.4260 | 0.1571 | 0.3978 | 0.1627 | 0.3898 | 0.1926 | 0.3737 | 0.2401 |
| WAPAM-LRec | 0.4230 | 0.1878 | 0.3903 | 0.1928 | 0.3827 | 0.2335 | 0.3614 | 0.2759 |

*Continued on next page ...*

| | 1-HTER | | HBLEU | | Adequacy | | Fluency | |
|---|---|---|---|---|---|---|---|---|
| | RMSE | r | RMSE | r | RMSE | r | RMSE | r |
| WAPAM-LF$_1$ | 0.4247 | 0.1784 | 0.3903 | 0.1835 | 0.3839 | 0.2225 | 0.3586 | 0.2688 |
| LTPAM-LPrec | 0.3729 | 0.1674 | 0.3983 | 0.1534 | 0.3700 | 0.1523 | 0.4374 | 0.1509 |
| LTPAM-LRec | 0.3646 | 0.2245 | 0.3822 | 0.2110 | 0.3540 | 0.2281 | 0.4154 | 0.2208 |
| LTPAM-LF$_1$ | 0.3607 | 0.2089 | 0.3758 | 0.1942 | 0.3498 | 0.2045 | 0.4066 | 0.2012 |
| LTPAM$_f$-LPrec | 0.3886 | 0.1837 | 0.3951 | 0.1699 | 0.3756 | 0.1739 | 0.4156 | 0.1841 |
| LTPAM$_f$-LRec | 0.3830 | 0.2323 | 0.3814 | 0.2221 | 0.3633 | 0.2384 | 0.3963 | 0.2445 |
| LTPAM$_f$-LF$_1$ | 0.3804 | 0.2190 | 0.3766 | 0.2061 | 0.3603 | 0.2183 | 0.3885 | 0.2281 |
| WTPAM-LPrec | 0.4166 | 0.1384 | 0.4013 | 0.1253 | 0.3886 | 0.1513 | 0.3928 | 0.1707 |
| WTPAM-LRec | 0.4122 | 0.1910 | 0.3896 | 0.1794 | 0.3782 | 0.2184 | 0.3747 | 0.2326 |
| WTPAM-LF$_1$ | 0.4131 | 0.1736 | 0.3889 | 0.1602 | 0.3792 | 0.1952 | 0.3711 | 0.2141 |
| WTPAM$_f$-LPrec | 0.4543 | 0.1465 | 0.4217 | 0.1336 | 0.4186 | 0.1636 | 0.3924 | 0.1889 |
| WTPAM$_f$-LRec | 0.4511 | 0.1886 | 0.4125 | 0.1776 | 0.4106 | 0.2185 | 0.3777 | 0.2383 |
| WTPAM$_f$-LF$_1$ | 0.4523 | 0.1745 | 0.4120 | 0.1620 | 0.4116 | 0.1999 | 0.3742 | 0.2247 |
| SeHC$_{pam}$ | 0.2414 | 0.2292 | 0.2833 | 0.2195 | 0.2414 | 0.2787 | 0.2661 | 0.3210 |
| SeHC$_{+pam}$ | 0.2445 | 0.2387 | 0.2822 | 0.2368 | 0.2370 | 0.3571 | 0.2575 | 0.3908 |
| B+SeHC$_{pam}$ | 0.2274 | 0.3977 | 0.2666 | 0.4069 | 0.2198 | 0.4854 | 0.2419 | 0.5016 |
| B+SeHC$_{+pam}$ | 0.2337 | 0.3417 | 0.2701 | 0.3697 | 0.2224 | 0.4694 | 0.2439 | 0.4881 |
| SeQE | 0.2249 | 0.3884 | 0.2710 | 0.3648 | 0.2242 | 0.4447 | 0.2404 | 0.5182 |
| B+SeQE | 0.2219 | 0.4194 | 0.2670 | 0.3975 | 0.2188 | 0.4882 | 0.2362 | 0.5427 |
| SSTK$_{new}$ | 0.2224 | 0.4152 | 0.2692 | 0.3788 | 0.2246 | 0.4409 | 0.2404 | 0.5158 |
| SeHC$_{new}$ | 0.2449 | 0.2278 | 0.2873 | 0.2083 | 0.2459 | 0.3077 | 0.2651 | 0.3550 |
| SeQE$_{new}$ | 0.2214 | 0.4223 | 0.2680 | 0.3892 | 0.2224 | 0.4584 | 0.2390 | 0.5233 |
| | Syntactico-semantic QE systems | | | | | | | |
| SSQE | 0.2246 | 0.3920 | 0.2696 | 0.3768 | 0.2230 | 0.4538 | 0.2402 | 0.5196 |
| B+SSQE | 0.2225 | 0.4144 | 0.2673 | 0.3953 | 0.2202 | 0.4771 | 0.2379 | 0.5331 |

Table A.2: Quality estimation results using the News data set

| | BLEU | | 1-TER | | Meteor | |
|---|---|---|---|---|---|---|
| | RMSE | r | RMSE | r | RMSE | r |
| | Baseline QE systems | | | | | |
| B-Mean | 0.1626 | - | 0.1965 | - | 0.1657 | - |
| B-WMT | 0.1601 | 0.1766 | 0.1949 | 0.1565 | 0.1625 | 0.2047 |
| | Syntax-based QE systems | | | | | |
| SyTK | 0.1581 | 0.2437 | 0.1888 | 0.2774 | 0.1595 | 0.2715 |
| B+SyTK | 0.1570 | 0.2696 | 0.1879 | 0.2939 | 0.1576 | 0.3111 |
| SyHC-all | 0.1603 | 0.2108 | 0.1902 | 0.2510 | 0.1607 | 0.2493 |
| SyHC | 0.1603 | 0.1998 | 0.1913 | 0.2365 | 0.1610 | 0.2516 |
| B+SyHC | 0.1587 | 0.2418 | 0.1899 | 0.2611 | 0.1585 | 0.2964 |
| SyQE | 0.1577 | 0.2535 | 0.1887 | 0.2797 | 0.1594 | 0.2743 |
| B+SyQE | 0.1568 | 0.2802 | 0.1879 | 0.2937 | 0.1576 | 0.3127 |
| SyQE$_L$ | 0.1583 | 0.2341 | 0.1887 | 0.2796 | 0.1600 | 0.2606 |
| SyTK$_L$ | 0.1583 | 0.2350 | 0.1888 | 0.2792 | 0.1600 | 0.2620 |
| SyHC$_L$ | 0.1609 | 0.1750 | 0.1914 | 0.2262 | 0.1613 | 0.2336 |
| SyTK/C-ST$_H$ | 0.1584 | 0.2307 | 0.1896 | 0.2641 | 0.1594 | 0.2748 |
| SyTK/C-ST$_L$ | 0.1582 | 0.2348 | 0.1890 | 0.2733 | 0.1596 | 0.2698 |
| SyTK/D-ST$_H$ | 0.1591 | 0.2103 | 0.1907 | 0.2412 | 0.1616 | 0.2213 |
| SyTK/D-ST$_L$ | 0.1597 | 0.1902 | 0.1913 | 0.2279 | 0.1623 | 0.2025 |
| SyTK/C-S$_H$ | 0.1583 | 0.2312 | 0.1904 | 0.2521 | 0.1590 | 0.2824 |
| SyTK/C-S$_L$ | 0.1582 | 0.2335 | 0.1901 | 0.2554 | 0.1599 | 0.2638 |
| SyTK/C-T$_H$ | 0.1608 | 0.1479 | 0.1925 | 0.2018 | 0.1620 | 0.2124 |
| SyTK/C-T$_L$ | 0.1616 | 0.1204 | 0.1934 | 0.1800 | 0.1632 | 0.1773 |
| SyTK/D-S$_H$ | 0.1598 | 0.1869 | 0.1925 | 0.2004 | 0.1630 | 0.1832 |
| SyTK/D-S$_L$ | 0.1601 | 0.1780 | 0.1933 | 0.1816 | 0.1630 | 0.1835 |
| SyTK/D-T$_H$ | 0.1598 | 0.2102 | 0.1916 | 0.2204 | 0.1622 | 0.2051 |
| SyTK/D-T$_L$ | 0.1604 | 0.1679 | 0.1924 | 0.2037 | 0.1628 | 0.1867 |
| SyHC/C-ST$_H$ | 0.1604 | 0.2046 | 0.1906 | 0.2443 | 0.1604 | 0.2538 |
| SyHC/C-ST$_L$ | 0.1613 | 0.1739 | 0.1894 | 0.2663 | 0.1599 | 0.2643 |
| SyHC/D-ST$_H$ | 0.1617 | 0.1593 | 0.1956 | 0.1609 | 0.1634 | 0.1813 |
| SyHC/D-ST$_L$ | 0.1125 | 0.1633 | 0.1944 | 0.1491 | 0.1638 | 0.1535 |
| SyHC/C-S$_H$ | 0.1613 | 0.1748 | 0.1930 | 0.1874 | 0.1621 | 0.2115 |
| SyHC/C-S$_L$ | 0.1616 | 0.1652 | 0.1933 | 0.1803 | 0.1620 | 0.2113 |
| SyHC/C-T$_H$ | 0.1622 | 0.1585 | 0.1936 | 0.1801 | 0.1622 | 0.2116 |
| SyHC/C-T$_L$ | 0.1624 | 0.1409 | 0.1935 | 0.1747 | 0.1630 | 0.1861 |
| SyHC/D-S$_H$ | 0.1385 | 0.1624 | 0.1939 | 0.1616 | 0.1626 | 0.1932 |
| SyHC/D-S$_L$ | 0.1381 | 0.1625 | 0.1946 | 0.1466 | 0.1636 | 0.1597 |
| SyHC/D-T$_H$ | 0.1643 | 0.0583 | 0.1983 | 0.0247 | 0.1659 | 0.0979 |
| SyHC/D-T$_L$ | 0.1720 | -0.0282 | 0.1978 | -0.0032 | 0.1655 | 0.0567 |
| SyTK/CD-S | 0.1584 | 0.2294 | 0.1899 | 0.2573 | 0.1596 | 0.2690 |
| SyTK/CD-T | 0.1597 | 0.2101 | 0.1913 | 0.2270 | 0.1613 | 0.2299 |
| SyTK/C-T$_m$ | 0.1591 | 0.2143 | 0.1940 | 0.1700 | 0.1602 | 0.2580 |
| SyTK/CD-ST$_m$ | 0.1574 | 0.2609 | 0.1880 | 0.2918 | 0.1588 | 0.2862 |