Quality Estimation of English-French Machine Translation: A Detailed Study of the Role of Syntax

Rasoul Kaljahi[†‡], **Jennifer Foster**[†], **Raphael Rubino**[†‡], **Johann Roturier**[‡]
[†]NCLT, School of Computing, Dublin City University, Ireland
{rkaljahi, jfoster, rrubino}@computing.dcu.ie
[‡]Symantec Research Labs, Dublin, Ireland
johann_roturier@symantec.com

Abstract

We investigate the usefulness of syntactic knowledge in estimating the quality of English-French translations. We find that dependency and constituency tree kernels perform well but the error rate can be further reduced when these are combined with hand-crafted syntactic features. Both types of syntactic features provide information which is complementary to tried-and-tested non-syntactic features. We then compare source and target syntax and find that the use of parse trees of machine translated sentences does not affect the performance of quality estimation nor does the intrinsic accuracy of the parser itself. However, the relatively flat structure of the French Treebank does appear to have an adverse effect, and this is significantly improved by simple transformations of the French trees. Finally, we provide further evidence of the usefulness of these transformations by applying them in a separate task – parser accuracy prediction.

1 Introduction

Quality Estimation (QE) for Machine Translation (MT) involves judging the correctness of the output of an MT system given an input and no reference translation (Blatz et al., 2003; Ueffing et al., 2003; Specia et al., 2009). An accurate QE-for-MT system would mean that reliable decisions could be made regarding whether to publish a machine translation as is or to re-direct it to a translator, either for post-editing or to be translated from scratch. The scores produced by a QE system can also be used to choose between translations, in a system combination framework or in n-best list reranking. The work presented here takes place in the context of a wider study, the aim of which is to develop an English-French QE system so that technical support material that is produced on a daily basis by a company's English-speaking customers can be translated automatically into French and made available with confidence to the company's French-speaking customer base.

It is reasonable to assume that syntactic features are useful in QE for MT 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 English-Spanish QE (Hardmeier et al., 2012; Rubino et al., 2012). We focus more closely on understanding the role of syntax by comparing the use of hand-crafted features and tree kernels (Collins and Duffy, 2002; Moschitti, 2006), and by teasing apart the contribution of target and source syntax.

We find that both tree kernels and manually engineered features produce statistically significantly better results than a strong set of non-syntactic features provided as a baseline by the organisers of the 2012 WMT shared task on QE for MT (Callison-Burch et al., 2012), and that both types of syntactic features can be combined fruitfully with this baseline. Furthermore, we show that it is worthwhile to combine tree kernels with hand-crafted features. Our tree kernel features are the complete set of tree fragments of both the constituency and dependency trees of the source and target sentences. Our hand-crafted feature set consists of an initial set of 489 constituency and dependency features which are then reduced to a set of 144 with no significant loss in performance.

Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, pages 2052–2063, Dublin, Ireland, August 23-29 2014.

We then show that source (English) constituency trees significantly outperform target (French) translation constituency trees in this task. We hypothesise that this is happening because a) the French parser has a lower accuracy compared to the English, or b) the target trees sentences are harder to parse, representing, as they do, potentially ill-formed machine translations which may result in noisier parse trees which are harder to learn from. If the first hypothesis were true, we would expect to see a drop in the accuracy of our QE system when we use lower-accuracy parses. We do not observe this. If the second hypothesis were true, we would expect to observe that the target trees were also less useful than the source trees in the opposite translation direction (French-English). Instead, we find that the target (English) constituency trees significantly outperform the source (French) constituency trees, suggesting that the difference between source and target that we observe in the original English-French experiment is related neither to intrinsic parser accuracy nor to translation direction but rather to the languages/treebanks.

We explore the extent to which the difference between French and English constituency trees is due to the relatively flatter structure of the French treebank. We use simple transformation heuristics to introduce more nodes into the French trees and significantly improve the performance. We also apply these heuristics in a second task, parser accuracy prediction. This task is similar to QE for MT except we are predicting the quality of a parse tree in the absence of a reference parse tree. We also find here that the modified trees also outperform the original trees, suggesting that one must proceed with caution when using French Treebank tree fragments in a machine-learning task.

The paper's novel contributions are as follows:

1. Evidence that syntactic information is useful in English-French QE for MT and further evidence that it is useful in QE for MT in general
2. A comparison of two methods of representing syntactic information in QE
3. A more comprehensive set of syntactic features than has been previously been used in QE for MT
4. A comparison of the role of source and target syntax in English-French QE for MT
5. A set of heuristics that can be applied to French Treebank trees resulting in performance improvements in the tasks of both QE for MT and parser accuracy prediction

The rest of this paper is organised as follows: we discuss related work in using syntax in QE in Section 2, we describe the data in Section 3, and we then go on to describe the QE framework and the systems built in Section 4. We follow this with an investigation of the role of source and target syntax in Section 5 before presenting our heuristics to modify the French constituency trees in Section 6.

2 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 the source sentence could be found. Hardmeier et al. (2012) employ tree kernels to predict the 1-to-5 post-editing cost of a machine-translated sentence. They use tree kernels derived from syntactic constituency and dependency trees of the source side (English) and only dependency trees of the translation side (Spanish). The tree kernels are used both alone and combined with non-syntactic features. The combined setting ranked second in the 2012 shared task on QE for MT (Callison-Burch et al., 2012). Rubino et al. (2012) explore a variety of syntactic features extracted from the output of both a hand-crafted broad-coverage grammar/parser and a statistical constituency parser on the WMT 2012 data set. They find that the syntactic features make an important contribution to the overall system. In a framework for combining QE and automatic metrics to evaluate MT output, Specia and Giménez (2010) use part-of-speech (POS) tag language model probabilities of the MT output 3-grams as features for QE and features built upon syntactic chunks, dependencies and constituent structure to build automatic MT evaluation metrics. Avramidis (2012) builds a series of models for estimating post-editing effort using syntactic features such as parse probabilities and syntactic label frequency. In a similar vein, Gamon et al. (2005) use POS tag trigrams, CFG rules and features derived from a semantic analysis of the MT output to classify it as fluent or disfluent.

In this work, we compare the use of tree kernels and hand-crafted features extracted from the constituency and dependency trees of the source and target sides of a translation pair, as well as comparing the role of source and target syntax. In addition, we conduct a more in-depth analysis of these approaches and compare the utility of syntactic information extracted from the source side and target sides of the translation.

3 Data

While there is evidence to suggest that predicting human evaluation scores is superior to predicting automatic metrics in QE for ME (Quirk, 2004), it has also been shown that human judgements are not necessarily consistent (Snover et al., 2006). A more practical consideration is that human evaluation exists for just a few language pairs and domains. To the best of our knowledge, the only available English-to-French data set which contains human judgements of translation quality 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[1], 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.

Thus, we instead attempt to predict automatic metric scores as there is a sufficient amount of parallel text for our language pair and domain. We use `BLEU`[2](Papineni et al., 2002), `TER`[3](Snover et al., 2006) and `METEOR`[4] (Denkowski and Lavie, 2011), which are the most-widely used MT evaluation metrics. All metrics are applied at the segment level.[5]

We randomly select 4500 parallel segments from the News development data sets released for the WMT13 translation task (Bojar et al., 2013). 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[6]: a phrase-based Moses system trained on training sets of WMT12 releases of Europarl and News Commentary plus data from Translators Without Borders (TWB)
- SYSTRAN: a proprietary rule-based system
- Bing[7]: an online translation system

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 analyses purposes.

4 Syntax-based QE

One way to employ syntactic information in a machine-learning task is to manually compile a set of features that can be extracted automatically from a parse tree. An example of one such feature is the label of the root of the tree. Another method is to directly use these trees in a *tree kernel* (Collins and Duffy, 2002; Moschitti, 2006). This approach allows exponentially-sized feature spaces (e.g. all subtrees

[1] http://www.faust-fp7.eu/faust/Main/DataReleases

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

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

[4] METEOR 1.4: http://www.cs.cmu.edu/~alavie/METEOR/

[5] We present `1-TER` to be more easily comparable to `BLEU` and `METEOR`. 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-TER`.

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

[7] http://www.bing.com/translator

of a tree) to be efficiently modelled using dynamic programming 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 QE for MT (Hardmeier et al., 2012). Although there can be overlap between the information captured by the two approaches, each can capture information that the other one cannot. In addition, while tree kernels involve minimal feature engineering, hand-crafted features offer more flexibility. Moschitti (2006) shows that combining the two is beneficial. We use both hand-crafted features and tree kernels, applied separately and combined together.

For parsing the English and French data into their constituency structures, a PCFG-LA parser[8] is used. 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). We obtain dependency parses by converting the English constituency parses using the Stanford converter (de Marneffe and Manning, 2008) and the French parses using Const2Dep (Candito et al., 2010). We evaluate the performance of the QE models using Root Mean Square Error (RMSE) and Pearson correlation coefficient (r). To compute the statistical significance of the performance differences between QE models, we use paired bootstrap resampling following Koehn (2004). We randomly resample (with replacement) a set of N instances from the predictions of each of the two given systems, where N is the size of the test 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 build using tree kernels, hand-crafted features and a combination of both.

4.1 Baseline QE Systems

In order to verify the usefulness of syntax-based QE, we build two baselines. The first baseline (BM) uses the mean of the segment-level evaluation scores in the training set for all instances. In the second baseline (BW), the 17 baseline features of the WMT12 QE Shared Task are used. BW 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[9] to build BW. The Radial Basis Function (RBF) kernel is used. The results for both baselines are presented in the first two rows of Table 1. Since BW is a stronger baseline than BM, we will compare all syntax-based systems to BW only.

4.2 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 benefitting 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[10], 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, omitting POS tag information. An example is shown in Figure 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.

Based on preliminary experiments on our development set, we use *subset* tree kernels, where the tree fragments are subtrees rooted at any node in the tree so that no production rule expanding a node in the

[8] https://github.com/CNGLdlab/LORG-Release. The Lorg parser is very similar to the Berkeley parser (Petrov et al., 2006), the main difference being its unknown word handling mechanism (Attia et al., 2010).

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

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

	BLEU		1-TER		METEOR	
	RMSE	r	RMSE	r	RMSE	r
BM	0.1626	0	0.1965	0	0.1657	0
BW	**0.1601**	**0.1766**	**0.1949**	**0.1565**	**0.1625**	**0.2047**
TK	0.1581	0.2437	0.1888	0.2774	0.1595	0.2715
BW+TK	**0.1570**	**0.2696**	**0.1879**	**0.2939**	**0.1576**	**0.3111**
HC	0.1603	0.1998	0.1913	0.2365	0.1610	0.2516
BW+HC	**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
BW+SyQE	**0.1568**	**0.2802**	**0.1879**	**0.2937**	**0.1576**	**0.3127**

Table 1: QE performances measured by RMSE and Pearson r; BM: Mean baseline, BW: WMT 17 baseline features, TK: tree kernels, HC: hand-crafted features, SyQE: full syntax-based systems (TK+HC). Statistically significantly better scores compared to their counterpart (upper row in the row block) are in bold.

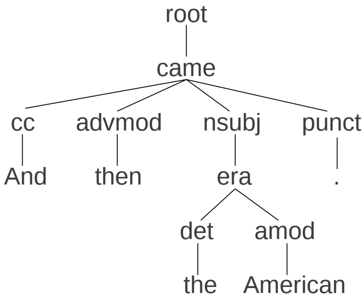

Figure 1: Tree Kernel Representation of Dependency Structure for *And then the American era came.*

subtree is split. Unlike *subtree* kernels, *subset* tree kernels allow tree fragments with non-terminals as leaves. We tune the C parameter for Pearson r on the development set, with all other parameters left as default.

We build a system with all four parse trees for every training instance, which includes the constituency and dependency trees of the source and target side of the translation. The third row of Table 1 shows the performance of this system which is named TK. The results achieved using this system represent a statistically significant improvement over the BW baseline results. In order to examine their complementarity, we combine these tree kernels and the baseline features (BW+TK) in the fourth row of Table 1. This combined system performs better than the two individual systems.

While BLEU prediction is the most accurate (lowest RMSE), METEOR prediction appears to be the easiest to learn (highest Pearson r). TER prediction seems to be more difficult than BLEU and METEOR prediction, especially in terms of prediction error. This is probably related to the distribution of each of these metric scores in our data set. The standard deviations (σ) of BLEU, TER and METEOR scores are 0.1620, 0.1943 and 0.1652 respectively. The substantially higher σ of TER scores makes them harder to predict accurately leading to higher prediction error.

4.3 Syntax-based QE with Hand-crafted Features

We design a set of constituency and dependency *feature types*, some of which have previously been used by the works described in Section 2 and some introduced here. Each feature type contains at least two features, one extracted from the source and the other from the translation. Numerical feature types can be further instantiated by extracting the ratio and differences between the source and target side feature values. Some feature types are parametric meaning that they can be varied by changing the value of a parameter. For example, the non-terminal label is a parameter for the non-terminal-label-count

	Constituency
$*1$	Label of the root node of the constituency tree
2	Height of the constituency tree which is the number of edges from root node to the farthest terminal (leaf) node
$*3$	Number of nodes in the constituency tree
4	Log probability of the constituency parse assigned by the parser
$*5$	Parseval F_1 score of the tree with respect to a tree produced by the Stanford parser (Klein and Manning, 2003)
$*6$	Right hand side of the CFG production rule expanding the root node
7	All non-lexical and lexical CFG production rules expanding the tree nodes
$*8$	Average arity of the non-lexical CFG production rules expanding the constituency tree nodes
9	Counts of each non-terminal label in the tree
$*10$	POS unigrams, 3-grams and 5-grams
11	POS n-gram scores against language models trained on the POS tags of the respective treebanks using the SRILM toolkit (`http://www.speech.sri.com/projects/srilm/`) with Witten-Bell smoothing
$*12$	Counts of each 12 universal POS tags (Petrov et al., 2012)
$*13$	Location of the first verb in the sentence in terms of the token distance from the beginning
$*14$	Average number of POS n-grams in each n-gram frequency quartile of the POS corpora of the respective treebanks
	Dependency
$*1$	POS tag of the top node (dependent of the dummy root node) of the dependency tree
$*2$	Number of dependents of the top node
$*3$	Sequence of all dependency relations which modify the top node
$*4$	Sequence of the POS tags of the dependents of the top node
$*5$	Average number of dependents per node
$*6$	Height of the tree computed in the same way for the constituency tree
$*7$	3- and 5-gram sequences of dependency relations of the tokens to their head
$*8$	Number of most frequent dependency relations in our `News` training set
$*9$	Dependency relation n-gram scores against language models trained on the respective treebanks for each language
$*10$	Average number of dependency relation n-grams in each n-gram frequency quartile of the respective treebanks
$*11$	Pairs of tokens and their dependency relations to their head

Table 2: Constituency and dependency feature types

feature type. Therefore, it instantiates as several features, one for each non-terminal-label.

As in `BW`, we use support vector machines (SVM) to build the QE systems using these hand-crafted features. We keep only those features which fire for more than a threshold which is set empirically on the development set. Table 2 lists our syntax-based feature types and their descriptions. Those that have, to the best of our knowledge, not been used in QE for MT before are marked with an asterisk.

The total number of feature-value pairs in the full feature set is 489. Since this feature set is large and contains many sparse features, we attempt to reduce it through ablation experiments in which we directly compare the effect of leaving out features that we suspect may be redundant. For example, we investigate whether either the ratio or difference of the source and target numerical features or both of them are redundant by building three systems, one without ratio features, one without difference features and one with neither. This process is also carried out for log probability and perplexity features, original and universal POS-tag-based features, n-gram and language model score features, lexical and non-lexical CFG rules, and n-gram orders (i.e. 3-gram vs. 5-gram features). This process proved useful: we found, for example, that either 3- or 5-grams worked better than both together and features based on universal POS tags better than those based on original POS tags.

The final reduced feature set contains 144 features-value pairs. We build one QE system with all 489 features `HC-all` and one with the reduced set of 144 features `HC` . Table 3 compares the performance on the development and test set. The system with the reduced feature set performs consistently better than the `HC-all` system on the development set, mostly with statistically significant differences. However, on the test set, the performance degrades albeit not statistically significantly. Considering a more than 70% reduction in feature set size, this relatively small degradation is tolerable. We use the reduced feature set as our hand-crafted feature set for the rest of the work.

Compared to `TK` in Table 1 (third and fourth versus fifth and sixth rows), the performances are lower for all MT metrics, though not statistically significantly. It is worth noting that we observed an opposite

	BLEU		1-TER		METEOR	
	RMSE	r	RMSE	r	RMSE	r
Development Set						
HC-all	0.1567	0.3026	0.1851	0.2746	0.1575	0.2996
HC	**0.1540**	0.3398	**0.1819**	**0.3263**	**0.1547**	0.3452
Test Set						
HC-all	0.1603	0.2108	0.1902	0.2510	0.1607	0.2493
HC	0.1603	0.1998	0.1913	0.2365	0.1610	0.2516

Table 3: QE performance with all hand-crafted syntactic features HC-all and the reduced feature set HC. Statistically significantly better scores compared to their counterpart (upper row) are in bold.

	RMSE	r
TK-CD-ST	0.1581	0.2437
TK-CD-S	0.1584	0.2294
TK-CD-T	0.1597	0.2101
TK-C-S	**0.1583**	**0.2312**
TK-C-T	0.1608	0.1479
TK-D-S	0.1598	0.1869
TK-D-T	0.1598	0.2102

Table 4: BLEU prediction performances with tree kernels of only source S or translation T side trees. The scores in bold are statistically better than their counterparts in the same row block. The original result with source and target combined is provided for reference in the first row.

behaviour on the development set, where hand-crafted features largely outperform tree kernels. This suggests that the tree kernels are more generalisable. We also combine these features with the WMT 17 baseline features (BW+HC). This combination also improves over both syntax-based and baseline systems, confirming again the usefulness of syntactic information in addition to surface features.

We combine tree kernels and hand-crafted features to build a full syntax-based QE system (SyQE), which improves over both TK and HC (Table 1) . The improvements for TER and METEOR prediction are slight but statistically significant for BLEU prediction. This system is also combined with BW in BW+SyQE (the last row of Table 1), resulting in statistically significant gains for all metrics.

5 Source and Target Syntax in Syntax-based QE

We now turn our attention to the parts played by source and target syntax in QE for MT. To save space, we present only the BLEU scores for the tree kernel systems. Table 4 shows the results achieved by systems built using either the source or target side of the translations.

At a glance, it can be seen that the source side constituency tree kernels outperform the target side ones, while the opposite is the case for dependency tree kernels. The differences for constituency trees are however substantially bigger. When both constituency and dependency trees are combined, the source side trees perform better (TK-CD-S vs. TK-CD-T).

The following three hypotheses could explain this difference between TK-C-S and TK-C-T:

1. **The Role of Parser Accuracy:** The fact that French parsing models do not reach the high Parseval F1s achieved by English parsing models could explain the difference in usefulness between the French and English consistuency trees. On the standard parsing test sets, the English parsing model achieves an F1 of 89.6 and the French an F_1 of 83.4.
2. **Parsing Machine Translation Output:** The difference between the source and target could be happening because the target side is machine translation output and (presumably) represents a lower

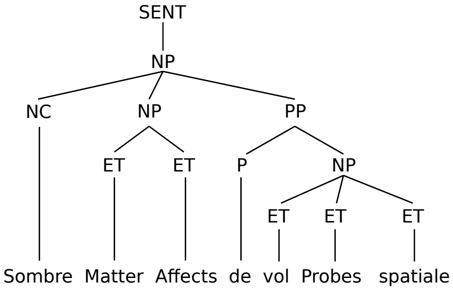

Figure 2: Parse tree of the machine translation of *Dark Matter Affects Flight of Space Probes* to French

quality set of sentences than the source (see Figure 2 for an example of a parse tree for a poor translation).

3. **Differences in Annotation Strategies:** The difference between the source and target could be due to the idiosyncrasies of the underlying treebanks which is not carried over via the conversion tools to the dependency structure.

Hypotheses 1 and 2 relate the usefulness of parse trees in QE to the intrinsic quality of the parse trees. French constituency trees are less accurate than English ones, either because the French parsing model is not as accurate as the English one (Hypothesis 1) or because the possibly ungrammatical nature of the French parsing input adversely affects the quality of the parse tree (Hypothesis 2). Although this low quality would be expected to affect the dependency trees in the same way since they are directly derived from the consistency trees, this is not the case and it appears that the problematic aspects of the French parses are abstracted away from the dependency trees.

To test the first hypothesis, we investigate the role of parser accuracy in QE. For both languages, we substitute the standard parsing models used in all our prior experiments with "lower-accuracy" models trained using only a fraction of the training data (following Quirk and Corston-Oliver (2006)). The English parsing model achieves an F_1 of 72.5 and the French an F_1 of 66.5, representing drops of approximately 17 points from the original models. The RMSE and Pearson r of the new QE model are 0.1583 and 0.2350 compared to 0.1581 and 0.2437 of the one trained with original trees (see also the third row of Table 1). These results show that the use of these lower-accuracy models has only a minimal and statistically insignificant effect on QE performance, suggesting that intrinsic parser accuracy is not the reason why the target constituency trees are less useful than the source constituency trees.[11]

To investigate the second hypothesis, 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. If the second hypothesis were true, the target side parse trees in this direction would still underperform the source side ones. The results are shown in Table 5. All the systems using target trees outperform those using source trees. The difference between source and target in the models that use constituency trees is especially substantial and statistically significant. Thus, it is apparent that the suspected lower quality of constituency parse trees of MT output is not the reason for the lower QE performance.

We now seek the answer in our third hypothesis, i.e. in the difference between the annotation schemes of the PTB and the FTB. One major difference, noted by, for example, Schluter and van Genabith (2007), is that the FTB has a relatively flatter structure. It lacks a verb phrase (`VP`) node and phrases modifying the verb are the sibling of the verb nucleus. We investigate this further in the next section.

6 Modifying French Parse Trees

In order to test 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 (`NP` or `PP`).

[11] See (Kaljahi et al., 2013) for a more detailed exploration of the role of parser accuracy in QE for MT.

	RMSE	r
TK-FE/CD-ST	0.1561	0.2334
TK-FE/CD-S	0.1574	0.1830
TK-FE/CD-T	0.1559	**0.2423**
TK-FE/C-S	0.1581	0.1578
TK-FE/C-T	**0.1556**	**0.2336**
TK-FE/D-S	0.1577	0.1655
TK-FE/D-T	0.1579	0.1886

Table 5: BLEU prediction performances with tree kernels for Fr-En direction (FE) (C: constituency, D: dependency, S: source, T: translation)

	RMSE	r
TK-C-T	0.1608	0.1479
TK-C-T$_m$	**0.1591**	**0.2143**
TK-CD-ST	0.1581	0.2437
TK-CD-ST$_m$	**0.1574**	**0.2609**

Table 6: QE with tree kernels using original and modified French trees ($_m$)

- *Heuristic 2* stratifies some of the production rules in the tree by grouping together every two equal 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 3 shows examples of the application of each of these methods. We apply these heuristics to the parsed MT output in the English-French translation direction and rebuild the tree kernel system with translation side constituency trees (TK-C-T) and the full tree kernel system (TK-CD-ST) with the modified trees. The results are presented in Table 6. Despite the possibility of introducing linguistic errors, these heuristics yield a statistically significant improvement in QE performance. Unsurprisingly, the changes are bigger for the system with only translation side constituency trees as in the full system there are three other tree types involved. These results suggest that the structure of the French constituency trees is a factor in the lower performance of its tree kernels in QE.[12]

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 3, 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.

The success of using modified French trees in improving tree kernel performance may of course depend on the data set and even the task in hand, and may not be generalisable. We next explore this question by applying the modification to a different task *and* a different data set.

6.1 Parser Accuracy Prediction

The task we choose is 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.

[12] 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.

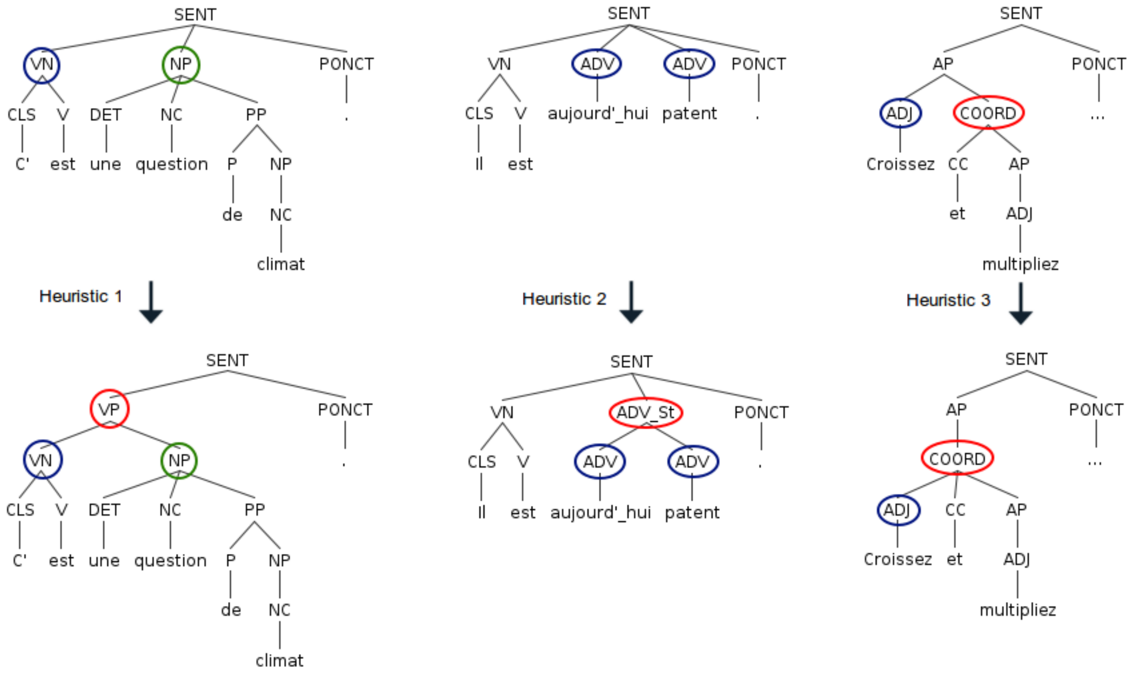

Figure 3: Application of tree modification heuristics on example French translation parse trees

	RMSE	r
PAP	0.1239	0.4035
PAP$_m$	0.1233	**0.4197**

Table 7: Parser Accuracy Prediction (PAP) performance with tree kernels using original and modified French trees ($_m$)

(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 7.

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.

7 Conclusion

We analysed the utility of syntactic information in QE of English-French MT and found it useful both individually and combined with standard QE features. We found that tree kernels are a convenient and effective way of encoding syntactic knowledge but that our hand-crafted feature set also brings additional, useful information. As a result of comparing the role of source and target syntax, we also found that the constituent structure in the FTB could be amended to be more useful in QE for MT and parser accuracy prediction. Now that we have explored the role of syntax in this project, our next step is try to further improve our QE system by adding semantic information. However, there are many other ways in which the research in this paper could be further extended. Our focus is on the language pair English-French and the QE task but it would certainly be interesting to perform a similar analysis on the role of syntax in QE for other language pairs, or to investigate the impact of French tree modification on other tasks.

Acknowledgments

This research has been supported by the Irish Research Council Enterprise Partnership Scheme (EPSPG/2011/102 and EPSPD/2011/135) and the computing infrastructure of the Centre for Next Generation Localisation at Dublin City University. We are grateful to Djamé Seddah for useful discussions about the French Treebank. We also thank the reviewers for their helpful comments.

References

Anne Abeillé, Lionel Clément, and François Toussenel. 2003. Building a treebank for French. In *Treebanks: Building and Using Syntactically Annotated Corpora*. Kluwer Academic Publishers.

Mohammed Attia, Jennifer Foster, Deirdre Hogan, Joseph Le Roux, Lamia Tounsi, and Josef van Genabith. 2010. Handling unknown words in statistical latent-variable parsing models for Arabic, English and French. In *Proceedings of the 1st Workshop on Statistical Parsing of Morphologically Rich Languages*.

Eleftherios Avramidis. 2012. Quality estimation for machine translation output using linguistic analysis and decoding features. In *Proceedings of WMT*.

John Blatz, Erin Fitzgerald, George Foster, Simona Gandrabur, Cyril Goutte, Alex Kulesza, Alberto Sanchis, and Nicola Ueffing. 2003. Confidence estimation for machine translation. In *JHU/CLSP Summer Workshop Final Report*.

Ondřej Bojar, Christian Buck, Chris Callison-Burch, Christian Federmann, Barry Haddow, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia. 2013. Findings of the 2013 workshop on statistical machine translation. In *Proceedings of the 8th WMT*.

Chris Callison-Burch, Cameron Fordyce, Philipp Koehn, Christof Monz, and Josh Schroeder. 2007. (meta-) evaluation of machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 136–158.

Chris Callison-Burch, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia. 2012. Findings of the 2012 workshop on statistical machine translation. In *Proceedings of the Seventh WMT*.

Marie Candito, Benot Crabb, and Pascal Denis. 2010. Statistical French dependency parsing: treebank conversion and first results. In *Proceedings of LREC*.

Michael Collins and Nigel Duffy. 2002. New ranking algorithms for parsing and tagging: Kernels over discrete structures, and the voted perceptron. In *Proceedings of the ACL*.

Marie-Catherine de Marneffe and Christopher D. Manning. 2008. The Stanford typed dependencies representation. In *Proceedings of the COLING Workshop on Cross-Framework and Cross-Domain Parser Evaluation*.

Michael Denkowski and Alon Lavie. 2011. Meteor 1.3: Automatic metric for reliable optimization and evaluation of machine translation systems. In *Proceedings of WMT*.

Michael Gamon, Anthony Aue, and Martine Smets. 2005. Sentence-level MT evaluation without reference translations: beyond language modeling. In *EAMT*.

Olivier Hamon, Antony Hartley, Andréi Popescu-Belis, and Khalid Choukri. 2007. Assessing human and automated quality judgments in the french MT evaluation campaign CESTA. In *Proceedings of the MT Summit*.

Christian Hardmeier, Joakim Nivre, and Jörg Tiedemann. 2012. Tree kernels for machine translation quality estimation. In *Proceedings of the WMT*.

Rasoul Samed Zadeh Kaljahi, Jennifer Foster, Raphael Rubino, Johann Roturier, and Fred Hollowood. 2013. Parser accuracy in quality estimation of machine translation: A tree kernel approach. In *Proceedings of IJCNLP*.

Dan Klein and Christopher D. Manning. 2003. Accurate Unlexicalized Parsing. In *Proceedings of the ACL*.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of EMNLP*.

Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Alessandro Moschitti. 2006. Making tree kernels practical for natural language learning. In *Proceedings of EACL*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the ACL*.

Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. Learning accurate, compact and interpretable tree annotation. In *Proceedings of the 21st COLING-ACL*.

Slav Petrov, Dipanjan Das, and Ryan McDonald. 2012. A universal part-of-speech tagset. In *Proceedings of LREC*.

Chris Quirk and Simon Corston-Oliver. 2006. The impact of parse quality on syntactically-informed statistical machine translation. In *Proceedings of EMNLP*.

Chris Quirk. 2004. Training a sentence-level machine translation confidence measure. In *Proceedings of LREC*.

Sujith Ravi, Kevin Knight, and Radu Soricut. 2008. Automatic prediction of parser accuracy. In *Proceedings of EMNLP*.

Raphael Rubino, Jennifer Foster, Joachim Wagner, Johann Roturier, Rasoul Kaljahi, and Fred Hollowood. 2012. DCU-Symantec submission for the WMT 2012 quality estimation task. In *Proceedings of WMT*.

Natalie Schluter and Josef van Genabith. 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*.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of AMTA*.

Lucia Specia and Jesús Giménez. 2010. Combining confidence estimation and reference-based metrics for segment level mt evaluation. In *Proceedings of AMTA*.

Lucia Specia, Marco Turchi, Nicola Cancedda, Marc Dymetman, and Nello Cristianini. 2009. Estimating the sentence-level quality of machine translation systems. In *EAMT*, pages 28–35.

Zhaopeng Tu, Yifan He, Jennifer Foster, Josef van Genabith, Qun Liu, and Shouxun Lin. 2012. Identifying high-impact sub-structures for convolution kernels in document-level sentiment classification. In *Proceedings of the ACL*.

Nicola Ueffing, Klaus Macherey, and Hermann Ney. 2003. Confidence measures for statistical machine translation. In *Machine Translation Summit IX*.

Michael Wiegand and Dietrich Klakow. 2010. Convolution kernels for opinion holder extraction. In *Proceedings of NAACL-HLT*.