A CCG-based Quality Estimation Metric for Statistical Machine Translation

Hala Almaghout
Center for Next Generation Localisation
Dublin City University
Dublin, Ireland
halmaghout@computing.dcu.ie

Lucia Specia
Department of Computer Science
University of Sheffield
Sheffield, United Kingdom
l.specia@sheffield.ac.uk

Abstract

We describe a metric for estimating the quality of Statistical Machine Translation (SMT) output based on syntactic features extracted using Combinatory Categorial Grammar (CCG). CCG has been demonstrated to be better suited to deal with SMT texts than context free phrase structure grammar formalisms. We use CCG features to estimate the grammaticality of the translations by dividing them into maximal grammatical chunks extracted from their CCG parse chart. We compare the performance of our CCG features with strong baseline and linguistic feature sets on French–English and Arabic–English data sets annotated with various quality scores. The results show that our CCG features outperform the baseline and linguistic features in most of the experiments. Furthermore, we demonstrate that our CCG features complement other types of features: combining CCG features with the baseline and other linguistic features furthers their performance.

1 Introduction

Quality Estimation (QE) for Machine Translation (MT) refers to the automatic prediction of translation quality without relying on reference translations. The availability of MT-translated data manually annotated with quality scores provides the ability to build machine learning systems which try to predict translation quality based on features extracted from the source sentence and/or its target translation and sometimes from internal translation information output by the MT system.

With the improvement of the quality of MT systems and their increasing use in real-world applications, MT QE has become increasingly more important. QE has been demonstrated to help in making the integration of MT systems in the translation pipeline more efficient. For example, using QE to filter out low-quality translations from the post-editing process has been shown to help in reducing post-editing time as low-quality translations might take more time to post-edit than to be translated from scratch (Specia, 2011). Furthermore, QE helps to enhance MT user experience by informing the user of the predicted quality of the translation produced by the MT system. Moreover, QE has been more and more used to enhance the quality of MT systems by integrating QE scores in n-best reranking and combining the translation of different MT systems.

QE features estimate the quality of the translation by capturing the aspects which evaluate translation quality, namely fluency and adequacy, in addition to predicting the difficulty of the translation. Adequacy refers to the extent to which the meaning of the source sentence is preserved by the translation, whereas fluency measures how grammatical the translation is. QE features based on language models have been usually used to evaluate the fluency of the translation. However, with the limited capacity of language model approaches to evaluating the grammaticality of the translation, approaches which use linguistic QE features extracted through syntactically and semantically analysing the translation output have emerged. Linguistic features extracted based on POS tagging, PCFG and dependency parsing the translation output have been integrated in QE mod-

Sima'an, K., Forcada, M.L., Grasmick, D., Depraetere, H., Way, A. (eds.)
Proceedings of the XIV Machine Translation Summit (Nice, September 2–6, 2013), p. 223–230.

els. One of the main challenges facing these approaches is that linguistic tools are trained on grammatical sentences only. Thus, using them to parse ungrammatical output can lead to inaccurate analyses. In this paper, we focus on predicting the fluency of the translation using linguistic features extracted based on Combinatory Categorial grammar (CCG). Thanks to CCG's flexible derivations and its rich syntactic categories, we were able to extract grammaticality QE features based on recognising grammatical chunks and examining sequences of CCG categories in the translation output. We also tackle the problem of parsing ungrammatical output by restricting the coverage of the CCG parser.

The rest of this paper is organised as follows. Section 2 reviews related work. Section 3 provides an introduction to CCG. Section 4 describes our approach. Section 5 presents our experiments. Finally, Section 6 concludes and provides avenues for future work.

2 Related Work

The first QE models were proposed by Blatz et al. (2004). They use data labeled with automatic MT metrics to learn QE models based on features extracted from the input and output sentences. Specia et al. (2009) add to the features proposed by Blatz et al. (2004) a set of features divided into "black-box" features i.e. MT system independent features and "glass-box" features i.e. features which use internal information from the MT system. They use training data annotated by both NIST and human annotation.

Using grammaticality features in QE has been demonstrated to improve their performance. Xiong et al. (2010) build a QE metric based on a Maximum Entropy classifier in which they integrate linguistic and lexical features to predict the correctness of each word in the translation output. Linguistic features are based on Link Grammar, which parses a sentence by pairing its words. They hypothesise that words which the parser fails to link to other words are likely to be grammatically incorrect. They demonstrate that linguistic features help to improve performance over lexical features and further improvement is gained when these two types are combined.

Avramidis et al. (2011) propose PCFG parsing-based QE features which represent the following information extracted from PCFG parse trees of the source and target sentences:

- Best parse tree log likelihood.
- Number of n-best trees.
- Confidence for the best parse tree.
- Average confidence of all trees.

Avramidis et al. (2011) demonstrate that these parsing-based features are able to achieve better correlation than non-linguistic-based features.

Specia et al. (2011) propose a set of QE features to predict the adequacy of translation. The features include the following syntactic features extracted from source and target dependency and constituency parse trees:

- Proportion of dependency relations with aligned constituents between source and target sentences.
- The same previous feature but with the order of constituents ignored.
- The same as the first feature but with Giza threshold equals to 0.1.
- Absolute difference between the depth of the syntactic tree for the source and the depth of the syntactic tree for the target.

Rubino et al. (2012) extract a set of syntax-based QE features originally developed to judge the grammaticality of sentences. Some syntactic features compare POS n-gram frequencies between the output sentence and a reference corpus. The features also include parsing features extracted from parse trees built using precision grammar, which is originally developed to detect grammatical errors. Other parsing-based features rely on information produced by parsers trained on well-formed and malformed sentences which result from introducing grammatical errors in the treebank on which the parser is trained.

Felice and Specia (2012) compare the performance of a set of linguistic features extracted from source and target sentences constituency and dependency trees with the baseline system of the WMT 2012 evaluation campaign (Callison-Burch et al., 2012). Some of these features compare syntactic structures between source and target sentences whereas other features focus on detecting common grammatical errors committed by SMT systems. They show that the linguistic features alone were not able to outperform the baseline system. However, they show that following a proper selection procedure for linguistic features helps to boost their performance over the baseline system.

3 Combinatory Categorial Grammar

CCG (Steedman, 2000) is a grammar formalism which consists of a lexicon that pairs words with lexical categories (supertags, cf. Bangalore and Joshi (1999)) and a set of combinatory rules which specify how the categories are combined. A supertag is a rich syntactic description that specifies the local syntactic context of the word at the lexical level in the form of a set of arguments. CCG builds a parse tree for a sentence by combining CCG categories using a set of binary combinatory rules. Most of the CCG grammar is contained in the lexicon, which is why CCG has simpler rules compared to CFG productions.

CCG categories are divided into atomic and complex categories. Examples of atomic categories are S (sentence), N (noun), NP (noun phrase), etc. Complex categories such as S\NP and (S\NP)/NP are functions which specify the type and directionality of their arguments and results. For example, the supertag assigned to the verb *read* in the sentence *he reads* is S\NP, which means that the verb *read* needs a NP playing the role of the subject to its left to constitute a full sentence S. The same verb *read* is assigned a different supertag (S\NP)/NP in the sentence *he reads a book*. The supertag (S\NP)/NP denotes a transitive verb which needs a NP to its left playing the role of the subject and a NP to its right playing the role of the object in order to constitute a full sentence S.

4 Our Approach

4.1 Motivation

CCG has many unique qualities which made it an attractive grammar formalism to be incorporated into SMT systems (Hassan et al., 2007; Hassan et al., 2009; Almaghout et al., 2010; Almaghout et al., 2012) . These qualities can also be exploited in building a CCG-based QE metric which evaluates the grammaticality of the translation output. First, CCG allows for flexible structures thanks to its combinatory rules. Thus, it is possible to assign a CCG category to phrases which do not represent standard syntactic constituents. This is an important feature for SMT systems as SMT phrases are statistically extracted, and do not necessarily correspond to syntactic constituents. This same feature can also be used to detect grammatical chunks in the translation output, which helps to estimate its grammaticality. Second, CCG supertags present rich syntactic information at the lexical level about the dependents and local context of each word in the sentence. Therefore, CCG supertags reflect important information about the syntactic structure of the sentence without the need to build a full parse tree. This provides the ability to extract grammaticality features based on examining sequences of CCG supertags of the words of the translation output as in the work of Hassan et al. (2007). This can also be incorporated as a feature in a QE metric.

4.2 CCG-based QE

State-of-the-art SMT systems use the phrase, which is a continuous string of words, as the basic translation unit. Thus, each translation output by these systems consists of a set of non-overlapping phrases learnt from the word-aligned training corpus. The phrases themselves are grammatical as they are extracted from a grammatical text. However, joining these phrases by the SMT system does not necessarily produce a grammatical translation, especially that most SMT systems do not use linguistic knowledge in the translation process. Therefore, grammaticality measures can be built based on the number of grammatical chunks detected in the translation output. The simplest type of these measures is to count the number of phrases which constitute the translation output. We can further refine this grammaticality measure to be more accurate by trying to recognise grammatical chunks beyond phrase boundaries. CCG's flexible structures, which provide the ability to recognise nonstandard constituents, are valuable to perform such task. Thus, parsing the translation output with CCG and then extracting the grammatical constituents in the translation output from the parsing chart might help to predict the quality of the translation output. Nevertheless, simply running the CCG parser on the translation output to extract grammatical chunks is not good enough for a number of reasons. First, the parser is trained on grammatical sentences only. Therefore, using it to parse ungrammatical sentences would yield inaccurate supertags and parse trees. Furthermore, due to the high flexibility of CCG derivations and inaccurate supertags assigned to the words of an ungrammatical translation, the CCG parser will succeed to build full CCG parse trees for many ungrammatical sentences, which hinders the ability

Translation: it is precisely the entry of virtual operators on the mobile market that has led to a considerable reduction in prices , which were subsequently react to the traditional operators .

Maximal Phrases: it is precisely ||| the entry ||| of virtual ||| operators on ||| the mobile ||| market that ||| has led to a ||| considerable reduction in ||| prices , which ||| were subsequently ||| react ||| to the traditional ||| operators .

Supertagged Translation: it|NP is|(S[dcl]\NP)/NP precisely|(S\NP)\(S\NP) ||| the|NP[nb]/N entry|N ||| of|(NP\NP)/NP virtual|N/N ||| operators|N on|(NP\NP)/NP ||| the|NP[nb]/N mobile|N/N ||| market|N that|(NP\NP)/(S[dcl]\NP) ||| has|(S[dcl]\NP)/(S[pt]\NP) led|(S[pt]\NP)/PP to|PP/NP a|NP[nb]/N ||| considerable|N/N reduction|N in|(NP\NP)/NP ||| prices|N ,|,|, which|(NP\NP)/(S[dcl]\NP) ||| were|(S[dcl]\NP)/(S[pss]\NP) subsequently|(S\NP)\(S\NP) ||| react|(S[b]\NP)/PP ||| to|PP/NP the|NP[nb]/N traditional|N/N ||| operators|N .|.

Maximal CCG Constituents: it is precisely the entry of virtual operators on the mobile market that has led to a considerable reduction in prices ,|**NP** ||| which|**(NP\NP)/(S[dcl]\NP)** ||| were subsequently|**(S[dcl]\NP)/(S[pss]\NP)** ||| react to the traditional operators .|**S[b]\NP**

Features:

#Phrases	#Constituent	% Supertag Mismatches	%Category Mismatches
13	4	23%	33%

Figure 1: An example for CCG-based QE features extracted from a translation output.

to evaluate the grammaticality of these sentences.

To solve these problems, we try to limit the coverage of the CCG parser by restricting the supertags assigned to the words of the translation output to the ones co-occurred with their containing phrases in the training data. This is performed according to the following steps. First, the target side of the training corpus is supertagged with a CCG supertagger. Then, a phrase table is extracted from the source corpus and the CCG-supertagged target corpus according to word alignments learnt from plain source-target corpus. This CCG-augmented phrase table specifies for each source-target phrase pair (f, e) the sequence of supertags s assigned to the words of the target phrase e along with probability of the supertagged phrase given the source phrase $p(s, e|f)$. Afterwards, the translation output is divided into non-overlapping maximal phrases (minimum number of non-overlapping phrases) according to the phrase-table. Finally, each phrase is assigned the highest-probability supertag sequence $S = max\, p(s, e|f)$ according to the CCG-augmented phrase table. After supertagging the translation output, the CCG parser is used to build a parsing chart for the translation output.

We extract the following features which predict the quality of the translation output. The first feature is the minimum number of CCG constituents which span the translation output $K = min\ n$ where n is the the length of the highest-probability sequence of non-overlapping CCG categories $c_1,\ c_2, ...,\ c_n$ which span the translation output according to the CCG parsing chart. If the parser succeeds in building a full parse tree for the translation output, then $K = 1$. We hypothesise that these constituents approximate the maximal grammatical chunks in the translation output. Thus, the smaller the value of this feature is, the more grammatical the translation output might be. Figure 1 shows the maximal CCG constituents extracted from a translation output along with some CCG-based QE feature values for the same example. We can see that the output is composed of 13 phrases according to the phrase table. The CCG parser is able to detect 4 maximal CCG constituents in the output.

As a CCG supertag explicitly specifies the type and directionality of the arguments it expects, we can use this information to detect grammatical flaws by checking the agreement of each supertag argument(s) with its adjacent supertags in the translation output. Therefore our second QE feature measures the percentage of argument mismatches M in subsequent supertags in the translation output out of the total number of all subsequent supertags in the translation output: $P = \frac{M}{L-1}$ where L is the length of the sentence. This feature was originally proposed by Hassan et al. (2007) to be integrated in the PB-SMT model during the decoding process. For example, in Figure 1, the supertag (S[dcl]\NP)/NP of the word *is* has a matching left argument with the supertag NP of the word *it*.

Our third QE feature is the percentage of argument mismatches in the maximal CCG constituents $c_1,\ c_2, ...,\ c_K$ retrieved from the parsing chart out of the total number of subsequent

CCG constituents. This feature is similar to the previous feature with the difference is that the previous feature examines the supertag sequence at the word level whereas this feature examines the CCG category sequence at the constituent level. In Figure 1, the CCG category (NP\NP)/(S[dcl]\NP) of the second phrase has a matching left argument NP with the CCG category NP of the first phrase, which indicates that the combination of these two phrases is likely to be grammatical. By contrast, the categories of the phrases *were subsequently* and *react to the traditional operators* do not agree as the CCG category of the former expects a verb in the passive form to its right (S[pss]\NP), whereas the category of the latter is a bare infinitival verb phrase (S[b]\NP). This indicates that there might be a grammatical flaw when combining these two phrases with each other.

Our fourth QE feature is the 5-gram supertag language model log probability of the supertag sequence of the translation output. The language model is built from the supertagged target training corpus. The integration of this feature in the PB-SMT model was examined by Hassan et al. (2007). We also include 5-gram supertag language model perplexity of the supertag sequence of the translation output as the fifth feature in our system. In addition to the previous CCG-based features, we add a feature for the number of maximal phrases in the translation output. This sums up to 6 features we used in our experiments.

5 Experiments

5.1 Data

The data we used in our experiments is French–English and Arabic–English data. The French–English data is news data from the WMT 2010 evaluation campaign (Callison-Burch et al., 2010). The data consists of 2525 French sentences and their translation produced by the Moses Phrase-Based Decoder.[1] The data is annotated with three types of human annotation: post-editing effort, Human Translation Edit Rate (HTER) (Snover et al., 2006) and post-editing time. The Arabic–English data is also news data from the DARPA GALE project. The data consists of 2585 Arabic sentences and their translation produced by a state-of-the-art Phrase-Based SMT system and annotated with adequacy scores. We created five random splits for each data set, each of which takes 90% of the sentences for training and 10% of the sentences for testing. We used the Berkeley Parser[2] to extract the PCFG features. For CCG supertagging and parsing, we used the parser and supertagger from the C&C tools.[3]

5.2 Baseline Systems

We compared the performance of our CCG-based QE features on the French–English data with two baseline systems:

- A QE system which uses a set of 80 shallow and system-independent features extracted from source sentences and their translation (Specia and Farzindar, 2010).
- A QE system which uses a subset of 17 features from the 80 features used by the previous baseline system. These 17 features were used to build the baseline system in the quality estimation task in the WMT 2012 evaluation campaign (Callison-Burch et al., 2012).

For the French–English data, we also compare the performance of our CCG-based QE features with PCFG parsing-based features proposed by Avramidis et al. (2011) (cf. Section 2) applied on the source and target sentences in addition to the target sentence alone to maintain a fair comparison with our CCG-based features, which are extracted from the target sentence only.

For the Arabic–English data, we compare the performance of our CCG-based QE features with two baseline systems:

- A QE system which uses 122 system-independent fluency, adequacy and complexity features (Specia et al., 2011).
- A QE system which uses the same 17 features used as baseline QE system in WMT12.

We also compare the performance of our CCG-based QE features on the Arabic–English data with 4 linguistic features proposed by Specia et al. (2011) (cf. Section 2), which are a subset of the 122 baseline features.

We use epsilon-Support Vector Regression algorithm with radial basis kernel from the scikit-learn tool (Pedregosa et al., 2011) to learn our models.

5.3 Experimental Results

We measured the performance of the different systems using Root Mean Squared Error (RMSE) for each annotation type on each of the five random

[1] http://www.statmt.org/moses/

[2] http://code.google.com/p/berkeleyparser/

[3] http://svn.ask.it.usyd.edu.au/trac/candc

System	**RMSE**		
	Effort	**HTER**	**Time**
base17	0.6809 ± 0.0640	**0.1693 ± 0.0306**	0.6997 ± 0.0439
base80	0.7154 ± 0.0785	0.1815± 0.0279	0.7185 ± 0.0710
ccg	0.6750 ±0.0630	0.1716 ±0.0307	**0.6937 ± 0.0444**
ccg+base80	0.7182 ± 0.0646	0.1828 ± 0.0260	0.7122 ± 0.0685
ccg+base17	**0.6734 ± 0.0628**	0.1700 ± 0.0260	**0.6905 ± 0.0433**
Source and Target PCFG Features			
pcfg	0.6800 ± 0.0606	0.1741 ± 0.0317	0.7147 ± 0.0393
pcfg+base80	0.7194 ± 0.0812	0.1790± 0.0309	0.7198 ± 0.0685
pcfg+base17	**0.6740 ± 0.0622**	0.1732 ± 0.0302	**0.6950 ± 0.0470**
pcfg+ccg	0.6775 ± 0.0591	0.1720 ± 0.0304	**0.6825 ± 0.0437**
pcfg+ccg+base80	0.7219 ± 0.0718	0.1796 ±0.0263	0.7168 ± 0.0610
pcfg+ccg+base17	**0.6738 ± 0.0614**	0.1728 ± 0.0238	**0.6859 ± 0.0450**
Target PCFG Features			
pcfg	0.6828 ± 0.0549	0.1745 ± 0.0320	0.7125 ± 0.0383
pcfg+base80	0.7177 ± 0.0767	0.1820 ± 0.0280	0.7186 ± 0.0690
pcfg+base17	0.6785 ± 0.0622	0.1707± 0.0299	**0.6976 ± 0.0430**
pcfg+ccg	0.6797 ± 0.0606	0.1719 ± 0.0305	**0.6934 ± 0.0436**
pcfg+ccg+base80	0.7200 ± 0.0636	0.1824 ± 0.0276	0.7155 ± 0.0663
pcfg+ccg+base17	**0.6742 ± 0.0615**	0.1709 ±0.0239	**0.6894 ±0.0411**

Table 1: Average RMSE for each of the baseline, CCG-based and PCFG parsing-based QE systems on the five en-fr random test sets for effort, HTER and post-editing time scores. Boldface figures indicate feature sets that are not different from each other at a statistically significant level, but are significantly better than all others within a given type of score (paired t-test with p <0.05).

test sets we extracted from the data. We then calculated the average RMSE for all the five test sets. We also examined the performance of the combination of different types of features. The results for the baseline, CCG-based and linguistic QE features in addition to their combination on the French–English and Arabic–English data are illustrated in Tables 1 and 2 .

From Table 1, we can see that CCG features alone are able to outperform both the 17 and 80 baseline features for all annotation types, except for the HTER score, where the 17 baseline features achieve the best performance. Combining CCG features with the 17 baseline features helps to achieve significant improvement over each feature type individually, except for the HTER score. CCG features combined with the 17 baseline features achieve the best performance for the effort score. By contrast, combining CCG features with the 80 baseline features does not in general help to improve the performance. Comparing the performance of CCG features with PCFG features for all the scores, Table 1 shows that CCG features outperform both source and target and target only PCFG features. Combining CCG features with PCFG features achieves further improvement over the PCFG features but not over CCG features, except for the time score, where combining CCG features with PCFG source and target features achieves the best performance among all the systems. Furthermore, combining PCFG features with CCG features and the 17 baseline features does not demonstrate to achieve improvement over CCG features combined with the 17 baseline features except for the time score.

In general, the experimental results on the French–English data show that although CCG features are only 6 target-side features, they are powerful enough to boost the performance of the 17 baseline features, as combining CCG features with the 17 baseline features helps to significantly improve the performance over individual feature types. As CCG features are target-side features only, adding the 17 baseline features, which combine source and target-side features, help to grasp more aspects of translation quality with more focus on translation fluency due to the addition of CCG features. Finally, CCG features achieve better performance than PCFG parsing features applied both on the target side only and on target and source sides. We believe this is due to the fact that CCG features are more effective in predicting translation fluency than PCFG features as CCG features such as the minimum number of grammatical chunks in translation output directly capture grammatical flaws in the sentence and thus are more accurate in expressing translation fluency than PCFG parser statistics. Furthermore, as the

System	RMSE
base17	0.7738 ± 0.0089
base122	0.7908 ± 0.0156
dependency	0.7969 ± 0.0125
dependency+base17	0.7683 ± 0.0088
ccg	0.7845 ± 0.0120
ccg+base17	**0.7581 ± 0.0235**
ccg+base122	0.7800 ± 0.0043
ccg+dependency	0.7750 ± 0.0123
ccg+dependency+base17	**0.7637 ± 0.0076**

Table 2: Average RMSE for each of the baseline, CCG-based and dependency-based QE systems on the five ar-en random test sets annotated with adequacy score. Boldface figures indicate feature sets that are not different from each other at a statistically significant level, but are significantly better than all others (paired t-test with p <0.05).

CCG and PCFG parsers are trained on grammatical data only, our adaptation of the CCG parsing approach we used to deal with ungrammatical output seems to help in acquiring more accurate features. Although adding PCFG features to CCG features does not help to achieve better performance than CCG features alone for the HTER and effort scores, combining CCG features with PCFG features helps to achieve a remarkable performance for the time score. Taking into consideration the paired t-test for the top system against all other systems, we can see from Table 1 that for the effort score, combining the 17 baseline features with both the PCFG and CCG features helps to achieve the top performance . For the HTER score, the 17 baseline features alone achieve significant improvement compared with all other systems. Whereas for the time score, we can see that combining CCG features with PCFG features helps to boost their performance to the same level of the systems which combine PCFG features with the 17 baseline features.

For Arabic–English data set, Table 2 shows that CCG features outperform the 122 baseline features but they are not able to outperform the 17 baseline features. However, combining CCG features with the 17 baseline features achieves the best performance among all the systems. Moreover, the system which combines the CCG features with the 17 baseline features and the system which combines the CCG features with the 17 baseline features and dependency features significantly outperforms all other systems. This demonstrates again that CCG features complement the 17 baseline features very well and help to significantly boost their performance. Furthermore, combining CCG features with the 122 baseline features helps to improve the performance over each feature type. Table 2 also shows that CCG features achieve better performance than dependency features. This also might be due to the lack of dependency parsing adaptation to deal with ungrammatical output, which affects parsing accuracy. The table also shows that combining CCG features with dependency features helps to improve the performance over each feature type. This also demonstrates that dependency features, which examine the preservation of grammatical relations during translation, complement our CCG features, which focus on translation fluency.

6 Conclusion and Future Work

In this paper, we presented a QE metric based on linguistic features extracted using CCG. These features try to predict the fluency of the translation by extracting maximal grammatical chunks from the translation output and examining agreement of CCG category labels at the lexical and constituent levels. We also tackled the problem of parsing ungrammatical output by restricting the CCG supertags assigned to the words of the translation prior to CCG parsing to the ones occurred in the training data. We conducted experiments which compared the performance of our CCG features with strong baseline systems which use system independent features and with a set of linguistic features. Our experiments demonstrated that our CCG features achieved better performance than the baseline systems and the linguistic features in most of the experiments. Furthermore, our experimental results showed that combining CCG features with the baseline features and with other linguistic features helped to improve their performance, which indicates that CCG features help to predict important aspects of translation quality not treated by other feature types.

In future work, we plan to integrate more CCG features in our CCG-based QE metric such as features extracted from the internal information output by the CCG parser and supertagger. Furthermore, we plan to use our CCG analysis of the translation output to spot the parts of the translation that needs to be post-edited. Furthermore, the richness of CCG categories can be employed to identify the types of errors committed in the translation such as a missing verb or noun in addition to automatically correct them. We also plan to integrate CCG features in n-best reranking during SMT decoding and ranking translations output by different SMT systems.

References

Almaghout, H., J. Jiang, and A. Way. 2010. CCG augmented hierarchical phrase-based machine translation. In *Proceedings of the 7th International Workshop on Spoken Language Translation*, pages 211 – 218, Paris, France.

Almaghout, H., J. Jiang, and A. Way. 2012. Extending CCG-based syntactic constraints in hierarchical phrase-based SMT. In *Proceedings of the 16th conference of the European Association for Machine Translation*, pages 28–30, Trento, Italy.

Avramidis, Eleftherios, Maja Popovic, David Vilar Torres, and Aljoscha Burchardt. 2011. Evaluate with confidence estimation: Machine ranking of translation outputs using grammatical features. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 65–70, Edinburgh, UK.

Bangalore, S. and A. Joshi. 1999. Supertagging: An approach to almost parsing. *Computational Linguistics*, 25(2):237–265.

Blatz, John, Erin Fitzgerald, George Foster, Simona Gandrabur, Cyril Goutte, Alex Kulesza, Alberto Sanchis, and Nicola Ueffing. 2004. Confidence estimation for machine translation. In *Proceedings of the 20th international conference on Computational Linguistics*, Geneva, Switzerland.

Callison-Burch, Chris, Philipp Koehn, Christof Monz, Kay Peterson, Mark Przybocki, and Omar F. Zaidan. 2010. Findings of the joint workshop on statistical machine translation and metrics for machine translation. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 17–53, Uppsala, Sweden.

Callison-Burch, Chris, 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 Workshop on Statistical Machine Translation*, pages 10–51, Montréal, Canada.

Felice, Mariano and Lucia Specia. 2012. Linguistic features for quality estimation. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 96–103, Montréal, Canada.

Hardmeier, Christian. 2011. Improving machine translation quality prediction with syntactic tree kernels. In *Proceedings of the 15th conference of the European Association for Machine Translation*, pages 233–240.

Hassan, H., K. Sima'an, , and A. Way. 2007. Supertagged phrase-based statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, pages 288–295, Prague, Czech.

Hassan, H., K. Sima'an, and A. Way. 2009. A syntactified direct translation model with linear-time decoding. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1182–1191, Singapore.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Rubino, Raphael, Jennifer Foster, Joachim Wagner, Johann Roturier, Rasul Samad Zadeh Kaljahi, and Fred Hollowood. 2012. Dcu-symantec submission for the wmt 2012 quality estimation task. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 138–144, Montréal, Canada.

Snover, Matthew, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of Association for Machine Translation in the Americas*, pages 223–231, Cambridge, MA.

Specia, Lucia and Atefeh Farzindar. 2010. Estimating machine translation post-editing effort with hter. In *AMTA 2010 Workshop Bringing MT to the User: MT Research and the Translation Industry*.

Specia, Lucia, Marco Turchi, Zhuaran Wang, John Shawe-Taylor, and Craig Saunders. 2009. Improving the Confidence of Machine Translation Quality Estimates. In *Proceedings of MT Summit XII*.

Specia, Lucia, Najeh Hajlaoui, Catalina Hallett, and Wilker Aziz. 2011. Predicting machine translation adequacy. In *Proceedings of the 13th Machine Translation Summit*, pages 513–520, Xiamen, China.

Specia, Lucia. 2011. Exploiting objective annotations for measuring translation post-editing effort. *15th Conference of the European Association for Machine Translation*, pages 73–80.

Steedman, M. 2000. *The syntactic process*. MIT Press, Cambridge, MA, USA.

Xiong, Deyi, Min Zhang, and Haizhou Li. 2010. Error Detection for Statistical Machine Translation Using Linguistic Features. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 604–611, Uppsala, Sweden.