Using Syntactic and Semantic Information in the Evaluation of Corpus-based Machine Translation

- Work supported by the EAMT-

Cristina Vertan, Melania Duma

Natural Language Systems Division, University of Hamburg

1. Aim of the project

The aim of this project was to define a new automatic evaluation metric for machine translation, based on the output of a weighted constraint dependency parser, to test it and measure correlations with human judgements.

The work was performed from January 2012 until December 2012, while the disemination took place during the first 6 months of 2013.

This paper is organised as follows. Section 2 presents a short state of the art of evaluation measures, section 2 introduces the weight ed constraint dependency parser we used while section 3 is dedicated to the work performed. In section 4 we list the dissemination results and further perspectives. Section 7 deals with financial aspects of the project

2. Short state of the art

Research into machine translation evaluation aims at the development of a set of automatic methods that measure accurately the correctness of an output generated by a machine translation (MT) system. However, this task is a difficult one mainly because communication in natural language is very complex and sometimes ambiguous. At the moment, MT systems are not completely capable of capturing these characteristics of natural language, which has a direct impact on the quality of the generated output, thus making the problem of MT evaluation a complex one.

Automatic evaluation of MT systems is based on the existence of a set of references, created by a human annotator. By using an automatic method of evaluation a score is obtained based on the similarity between the output of the MT system and these references. The similarity can be calculated at different levels: lexical, syntactic or semantic. At the lexical level, the metrics developed so far can be divided into two major categories: n-gram based and edit distance based. From the category of n-gram based metrics one of the most popular methods of evaluation is BLEU (Papineni et al., 2001). It provides a score that is based on the summed number of n-grams shared by the references and the output, divided by the total number of n-grams. Lexical metrics based on edit distance are constructed using the Levenshtein distance applied at the word level. One of these metrics is WER (Nießen et al., 2000), which calculates the minimal number of insertion, substitutions and deletions needed to transform the candidate translation into a reference.

The main disadvantage of these metrics that are based on lexical matching is the fact that they do not take into account the variation that can be encountered in natural language. Thus they reward an otherwise fluent and syntactically correct candidate translation with a low score if it does not share a certain number of words with the set of references. Because of this, major disagreements between the scores awarded by BLEU and human judgments have been reported in (Koehn & Monz, 2006) and (Callison-Burch et al., 2006). Another disadvantage is that many of them cannot be used at segment level, which is often needed in order to better assess the quality of machine translation output and to determine which improvements should be made to the MT systems. Because of these disadvantages there is an increasing need for other approaches to MT evaluation that go beyond the lexical level of the phrases compared.

One of these approaches is the work described in (Liu and Gildea, 2005) which presents three evaluation metrics based on syntax and dependency trees. The first of these metrics, STM, is based on

determining the number of subtrees that can be found in both the candidate translation and the reference syntax trees. A kernel based subtree metric, TKM, is also introduced which is defined as the maximum of the cosine measure between the output and the set of references. The idea of syntactic similarity is further exploited in (Owczarzak et al., 2007) which uses a Lexical Functional Grammar (LFG) parser. The similarity between the translation and the reference is computed based on the precision and recall of the dependencies that describe the pair of sentences. Furthermore, paraphrases are used in order to improve the correlation with human judgements. A new set of syntactic metrics is also introduced in (Gimenez, 2008) and some of them are based on analysing different types of linguistic information (i.e. part-of-speech, lemma).

3. Weight Constraint Dependency Grammar

The goal of constraint dependency grammars (CDG) is to create a dependency structure that represents a given phrase (Schröder et al., 2000). These structures are often represented as trees due to the fact that no cycles can be present. A relation between two words in a sentence is represented by using an edge, which connects the regent and the dependent. Annotations that use different labels are assigned to the edges, in order to provide better distinction between the types of relations. The main advantage of using constraint dependency grammars over dependency grammars that use generative rules is that they can provide analysis of languages with a fragmented word order (Foth, 2004). An example of a dependency parse tree, obtained using the Weighted Constraint Dependency Grammar (WCDG) parser (Menzel & Schröder, 1998) is presented in Fig. 1. In this case, the labels are annotated with different syntactic functions and the analysis is performed on a syntactic level.

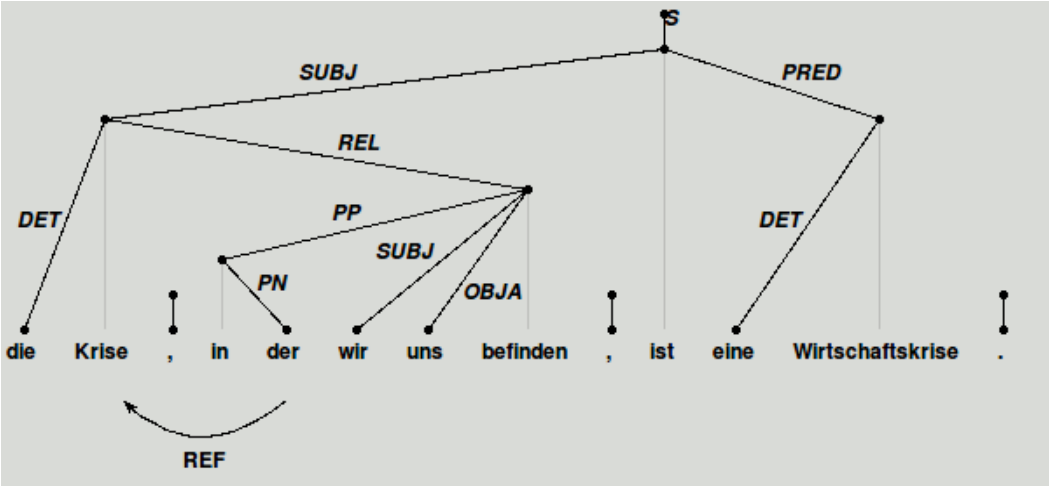


Figure 1 An example of a dependency parse tree

The idea behind WCDG is to assign different weights to the constraints that form the grammar. A constraint is made up of a logical formula that describes properties of the tree. One property that is always enforced is that no word can have more than one regent on any level at a time. Every constraint in WCDG is assigned a score which is a number between 0.0 and 1.0, where the general score of a parsing is calculated as the product of all the scores of all the instances of constraints that have not been satisfied. Rules that have a score of 0 are called hard rules, meaning that they cannot be ignored, which is the case of the one regent only rule mentioned earlier. The advantage of using graded constraints, as opposed to crisp ones, stems from the fact that they often act as a mean of mediation between different possible correct dependencies parse structures.

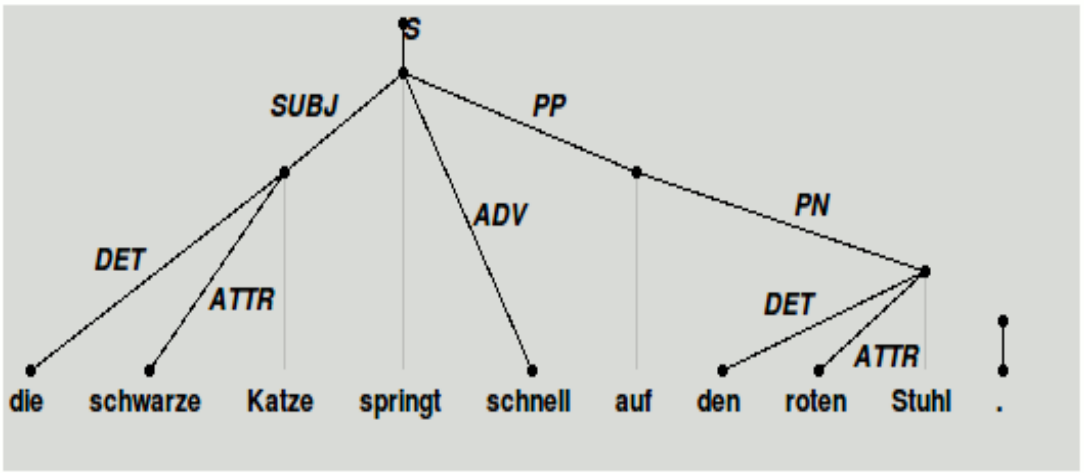
Each of the constraints is applied to every edge or every pair of edges from the dependency parse tree and in the case of the constraints that are violated, their scores are multiplied thus resulting in a score that represents the total score of the parsing.

The reason why we decided to concentrate on a dependency parser was because, as opposed to constituent parsers, it offers the possibility of better representing non-projective structures. Moreover, it has been shown (Kübler and Prokic, 2006) that in the case of German, the results achieved by a dependency parser are more accurate than the ones obtained when parsing using constituent parsers, and this is because dependency parsers can handle better long distance relations and coordination. The main advantage of using WCDG is that it is not restricted to well-formed input, which makes it interesting from the perspective of MT output. Because of the fact that the candidate translations are sometimes not well-formed, parsing them represents a challenge. However, WCDG will always provide a final result, in the form of a dependency structure, even though it might have a low score due to the violated constraints. Another advantage of using WCDG is that, in addition to the final score of the parsing, it provides information on the violated constraints, which can help perform error analysis.

3. Work performed

3.1. Definition of an Evaluation measure based on WCDG

The idea behind the new syntactic metric was to incorporate the WCDG parser in the process of evaluation. Because the end result of parsing with WCDG is a dependency tree, we have looked into techniques of measuring how similar two trees are. We wanted to determine whether a tree similarity metric applied on the two dependency parse trees would prove to be an efficient way of also capturing the similarity between the reference and the translation. Let us consider this example, in which the reference sentence is “Die schwarze Katze springt schnell auf den roten Stuhl.” and the candidate translation is“Auf den roten Stuhl schnell springt die schwarze Katze”. Even though the word order of the two segments is quite different, they manage to maintain the same meaning. We present in Figure 2 the dependency parse trees, obtained using WCDG, for the sentences considered. We can observe that the general structure of the translation is similar to that of the reference, the only difference being the reverse positions between the left subtree and the right subtree.



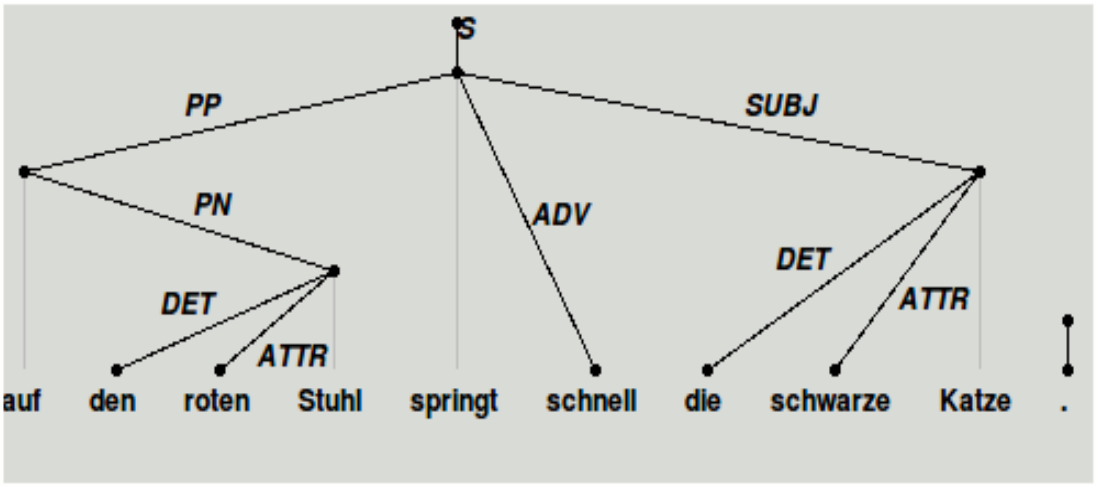


Figure 2 Example of dependency parse trees for reference and candidate translation

The output of the WCDG parser is presented in Figure 3. It is made up of chuncks, that describe the dependencies, that combined can be represented in the form of a dependency trees. It can be observed that WCDG provides also syntactical information about the tokens of the parsed phrase, information that can be integrated in the evaluation of machine translation. In order to apply the all common embedded subtree similarity, we first modified the dependency trees by removing the labels assigned to every edge, but maintaining the nodes and the left to rigth order between them. A question arised on whether to define the nodes as the actual tokens or using this syntactical information provided by WCDG. However, we have concluded that using only the syntax labels as nodes would strip away too much lexical information, which in turn would affect the performance of our proposed metric.

```
wordgraph4 <->
  0 1 auf
case / acc
cat / APPR
SYN -> PP -> 5 // springt
REF -> '' -> 0
,
  1 2 den
case / acc
cat / ART
gender / masc
number / sg
SYN -> DET -> 4 // Stuhl
REF -> '' -> 0
,
  2 3 roten
base / rot

,
```

```
case / acc
cat / ADJA
degree / positive
gender / masc
number / sg
SYN -> ATTR -> 4 // Stuhl
REF -> '' -> 0
,
  3 4 Stuhl
base / Stuhl
case / nom_dat_acc
cat / NN
gender / masc
number / sg
person / third
SYN -> PN -> 1 // auf
REF        ->        ''        ->        0
```

Figure 3: A fragment of the output of WCDG parser

The tree similarity measure that we chose to use was the All Common Embedded Subtrees (ACET) (Lin et al., 2008) similarity. An embedded subtree of a tree T is obtained by removing one or more nodes, which are not the root, from the tree T. The ACET similarity is defined as the number of common embedded subtrees shared between two trees.

In our experiments, we have applied the ACET algorithm presented in (Lin et al., 2008), and computed the number of common embedded subtrees between the dependency parse trees of the hypothesis and the reference. Because of all the additional information that parsing provides, like details about the syntactic characteristics, pre-processing of the output of the WCDG parser was necessary in order to transform the dependency tree into a general tree. We first removed the labels assigned to every edge, but maintained the nodes and the left to right order between them. An overview of the interaction between the components involved in the new metric is presented in Figure 4.

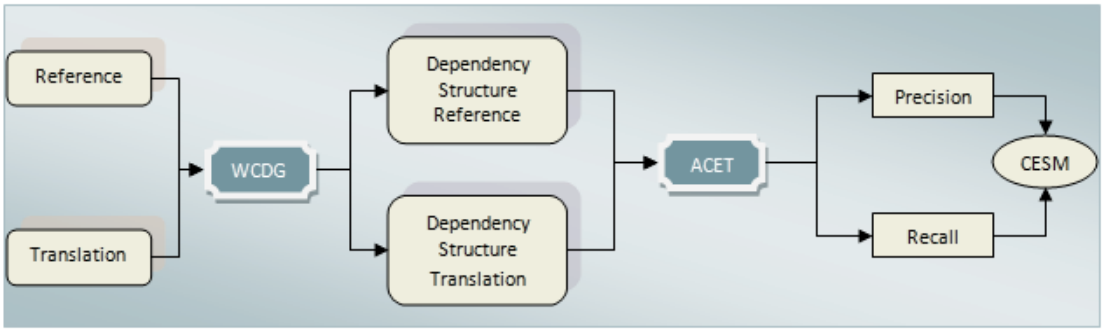


Figure 4 Interaction between components involved in computing CESM

The new metric proposed, which we decided to refer to as CESM (Common Embedded Subtree Metric), is based in the notion of precision, recall and F-measure of the common embedded subtrees of the reference and the translation. It is computed as follows, where $tree_{ref}$ and $tree_{hyp}$ represent the preprocessed dependency trees:

$$CESM = \frac{\sum common\ embedded\ subtrees\ (tree_{ref}, tree_{hyp})}{\sum common\ embedded\ subtrees\ (tree_{ref}, tree_{ref})}$$

3.2. Evaluation of the CESM Metric

In order to determine how well does CESM capture the similarity between references and translations, we evaluated it at system level and at segment level. The evaluation was conducted using data provided by the NAACL 2012 SMT workshop (Callison-Burch et al., 2012).

At system level, the initial German test set provided at the workshop was filtered according to the length of segments. As a result, 500 segments, with length between 50 and 80 characters, were extracted from the German reference file. In the next step, from the 15 systems that were submitted for evaluation in the English to German translation task, we selected 7 of them. After this initial step of filtering the data, we evaluated the outputs of the 7 systems. This was achieved by calculating the CESM score for every pair of reference and translation segments corresponding to a system. The average scores obtained by every system are depicted in Table 1. Evaluation of the metric at system level was performed by measuring the correlation of the CESM metric with human judgments using Spearman's rank correlation coefficient ρ. In order to compute the ρ score, the scores attributed to every system by CESM, were converted into ranks.

The ρ rank correlation coefficient was calculated as being ρ = 0.92.

No.	System name	Human rank	CESM rank	CESM score
1	DFKI	7	7	0.069
2	JHU	5	6	0.073
3	KIT	3	3	0.090
4	OnlineA	2	1	0.093
5	OnlineB	1	2	0.091
6	OnlineC	4	4	0.085
7	UK	6	5	0.075

Table 1: Ranks and scores assigned to the systems

The first step in evaluating at segment level was again filtering the initial test set provided by the NAACL workshop. Similarly to the system level evaluation, 500 segments were selected, that had the length between 50 and 80 characters. These 500 segments served as a template for the creation of the other files, one for every MT system. The Kendall tau rank correlation coefficient was calculated in order to measure the correlation with human judgments,

In order to calculate the value of Kendall tau, we determined the number of concordant pairs and the number of discordant pairs and the result was a correlation of 0.058. As a reference, the highest correlation for segment level reported in (Callinson-Burch et al, 2012) was 0.19 obtained by TerrorCat (Fishel et al., 2012) and the lowest was BlockErrCats (Popovic, 2012) with 0.040. The result obtained may be partially explained by the fact that only one judgment of a pair of reference and translation was taken into consideration. It will be interesting to decide in what way can the averaging of the ranks of a translation influence the correlation coefficient. Taking into consideration the results presented above, we can conclude that CESM has strong correlation with human judgments at system level and good correlation at segment level. In the future, the problem of improving even more the quality of CESM will be further explored.

One idea to improve the metric is in fact to optimize the parsing result, throughout a predictor. This is the subject of the next section

3.3. Implmentation of a Predictor componenent in order to optimize the measure

Our approach of improving parsing quality of MT output in the context of syntactic evaluation is based on integrating syntactical information that is extracted from the reference, into the processing of the translation. In order to implement this, we have designed a predictor component which we have integrated into WCDG, in order to facilitate the process of making informed decisions during the parsing of the translation. We decided to concentrate on a dependency parser because, as opposed to constituent parsers, it offers the possibility of better representing non-projective structures. Moreover, it has been shown (Kuebler and Prokic, 2006) that in the case of German, the results achieved by a dependency parser are more accurate than the ones obtained when parsing using constituent parsers, and this is because dependency parsers can handle better long distance relations and coordination.

The advantage of using the WCDG parser is that it gives further information on a parse, like the general score of the parse and the constraints that were violated during the parsing process. This information can be further explored in order to perform error analysis. Moreover, because of the fact that the candidate translations are sometimes not well-formed, parsing them represents a challenge. However, WCDG will always provide a final result, in the form of a dependency structure, even though it might have a low score due to the violated constraints.

By improving the parsing quality, we would also improve the accuracy of the general score of the parse, which could be further used as an automatic evaluation metric of translation quality. Moreover, the list of constraints violated by the translation could be additionally processed in order to perform error analysis.

3.3.1. Integration of a predictor component into the WCDG parser

The process of parsing using WCDG is based on three main important steps (McCrae, 2007). At first the sentence is pre-analyzed by a set of independent processing components, like a part-of-speech tagger. The next step is checking for satisfaction of grammar constraints, taking also into consideration the information provided by the pre-processing components. The last step is finding the dependency structure that has the lowest total penalty, where the penalty is the product of all the instances of constraints that had been violated. The new predictor component that we have designed is one of the external pre-processing components that initially analyzes the input, as can be observed in Figure 1. The main essential steps followed when parsing MT output using the predictor are:

1. Parse the reference using the WCDG parser and analyze the output, by extracting information about the dependency relations and ignoring the label information.
2. Based on these dependencies, create a mapping between the tokens of the reference and the translation. We highlight the fact that at this step, no parsing of the translation has been performed, therefore no syntactic information about the translation is available
3. Turn on the MT predictor component
4. Parse the translation, taking into consideration, at every step of the parsing process, the predictions made by the MT predictor. The predictions made by the MT predictor have the form of *token i has the regent token j* and are based on the mapping between the reference and candidate translation, and on the dependencies extracted during the first step.

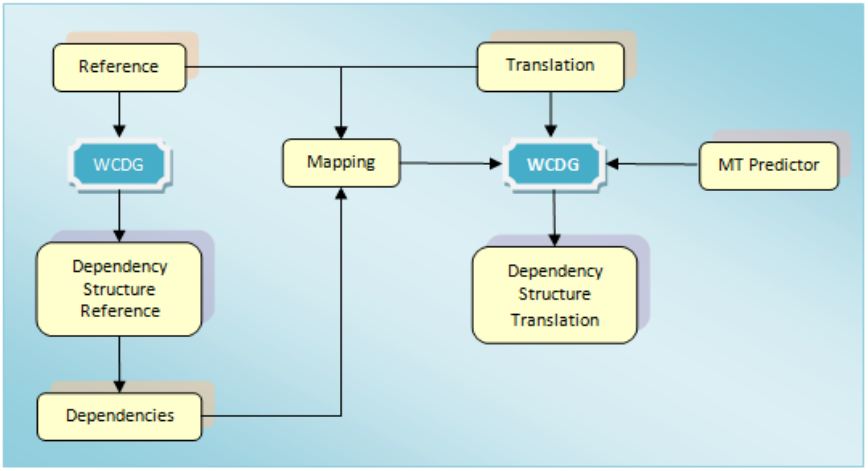


Figure 5: Integration of the predictor into the parsing of the reference and translation

Implementation of the new MT predictor component required alteration of the standard WCDG parser. A new pre-processing module was implemented that deals with the mapping between tokens belonging to the reference and those belonging to the translation. The mapping has been approached from a heuristic point of view and the designed algorithm is presented in Figure 2. The main idea behind the mapping algorithm is that the set of dependency relations will guide the process of mapping.

Therefore, for every dependency relation the regent and the determiner are calculated. Then every token of the translation is compared with the determiner and if a match is found then the token is mapped to the determiner. Next, every token in the vicinity of the matched one is compared with the regent and if a match is found then the new pair is also added to the mapping set. We chose to represent a vicinity of $token_i$ as being made up of the five tokens to the left of it and the five tokens to the right of it. In order to not penalize lexical variation we consider that

two tokens are matched, when they share the same stem and the same part-of-speech tag. The part-of-speech tags were obtained using the Stanford Log-linear Part-Of-Speech Tagger (Toutanova et al., 2003), while the stems were obtained using the German Snowball stemmer (Porter, 2001).

```
procedure Mapping
set ref = dependencies_reference;
      for every rd_k in ref
      Node regent=get_regent(rd_k);
      Node determiner = get_determiner(rd_k);
      Set tok=tokens_of_the_translation;
      for every token_i in tok
            if (token_i=determiner) then
             add (token_i,determiner) to the mapping set;
             for every token_j in the (i-5,i+5) interval
             if (token_j=regent) then
                  add (token_j,regent)to the mapping_set;
```

Figure 6: Mapping algorithm

The next step is to predict dependency relations for the tokens of the translation based on the mapping performed earlier. Predictions are made based on the algorithm presented in Figure 3. Every dependency is processed and the regent and the determiner are established. After this, a check is made to see if the regent and the determiner appear in the mapping set. If they do appear then the tokens, which we denoted using *regent_mapping* and *determiner_mapping*, are extracted from the mapping set. Finally, a prediction stating that the regent of token *determiner_mapping* should be token *regent_mapping* is added to the *prediction_map*. In the end, this prediction map will act as a guideline for the parsing of the translation and it should aide the process of making decision about the regent of a token.

```
procedure MTPredictor
set ref= dependencies_reference;
for every rd_k in ref
  Node regent=get_regent(rd_k);
  Node determiner=get_determiner(rd_k);
  if (regent is mapped in the translation and
  determiner is mapped in the translation)
   Node regent_mapping= mapping_set(regent);
   Node determiner_mapping = mapping_set
   (determiner);
   add prediction (regent_mapping,
   determiner_mapping) to the prediction_map;
```

Figure 7: Prediction algorithm

In the end, we also added a new constraint to the grammar, which is presented below. The constraint is verified for every edge on the syntax level of the dependency structure. The logical formula of the constraint translates to: "if a prediction has been made by the MT Predictor for the regent of edge X, then the regent appointed by WCDG must be the same as the regent indicated in the prediction". In the case that the constraint is violated, a penalty score of 0.5 will be acquired. This score allows the parser a certain independence in deciding whether a prediction is correct or not and offers the possibility of ignoring it in the case that it considers it to be incorrect.

3.3.2. The Headword Chain Metric

The evaluation of the predictor component was performed by using an automatic syntactic metric of evaluation of MT output. The metric that we have chosen was the HWCM metric (Liu and Gildea, 2005) which computes a score based on the number of matched n-grams of headword dependency chains. A headword dependency chain is defined as the sequence of words which forms a path in a tree

The headword chains are extracted using the recursive algorithm provided in (Liu and Gildea, 2005), which computes them in order from the shortest chain to the longest one. The chains, corresponding to the dependency tree depicted in Figure 2, are presented in Table 1.

Length	Headword Chains
1	*die*, *schwarze*, *Katze*, *springt*, *schnell*, *auf*, *den*, *roten*, *Stuhl*,
2	*Katze die, Katze schwarze, springt schnell, springt auf, auf Stuhl, Stuhl den, Stuhl roten*
3	*springt Katze die, springt Katze schwarze, springt auf Stuhl, auf Stuhl den, auf Stuhl roten*
4	*Springt auf Stuhl den, springt auf Stuhl roten*

Table 2: Examples of headword chains

Once extracted, the headword chains can be used to compute the HWCM score, which is defined as below, where *D* is the maximum length of the chain. While both *count(g)* and *count$_{clip}$(g)* represent the number of times that chain g appears in the dependency tree of the hypothesis, the latter cannot exceed the maximum number of times the chain occurs in the reference translation.

$$\mathrm{HWCM} = \frac{1}{D}\sum_{n=1}^{D}\frac{\sum_{g\in \mathrm{chain}_n(hyp)} \mathrm{count}_{\mathrm{clip}}(g)}{\sum_{g\in \mathrm{chain}_n(hyp)} \mathrm{count}(g)}$$

The evaluation performed in (Liu and Gildea, 2005) has shown that HWCM correlates better than BLEU with human judgments, which proves that HWCM is an appropriate method of evaluating MT quality.

3.4. Evaluation of the predictor Component

The HWCM metric was implemented and used in order to perform evaluation at system and segment level. Even though the initial metric performed lexical matching of the headword chains, we have altered it so that the chains were matched based on stemming (Porter, 2001) and part-of-speech tagging (Toutanova et al., 2003). We have chosen this approach because initial experiments have demonstrated that this modified metric captures better the improvements achieved by the predictor component. The predictor component was evaluated indirectly through the use of this syntactic metric. The reason why we have chosen this method of evaluation is because it provides an easier way of assessing the quality of parsing, as opposed to constructing correct dependency trees for ungrammatical sentences.

3.4.1. Experimental setup

The evaluation was conducted using data provided by the NAACL 2012 SMT workshop (Callison-Burch et al., 2012). The test data for the workshop is made up of 99 translated news articles in

English, German, French, Spanish and Czech. The workshop had an evaluation task, during which the data which was gathered during the translation task was used to evaluate the automatic methods that were submitted. The metrics had to provide both a score at system level and also a score at segment level. At system level, Spearman's rank coefficient was calculated, in order to measure how well do the metrics correlate with human judgments. At segment level, the correlation with human judgments was calculated using Kendall's tau rank correlation coefficient.

3.4.2. System level evaluation

At system level, the initial German test set provided at the workshop was filtered taking into account the length of the segments. As a result, 500 segments, with length between 50 and 140 characters, were extracted from the German reference file. The main reason for this filtering of data is because we also had to consider that the time needed to correctly analyze a sentence increases with the length of it. From the 15 systems that were submitted for evaluation in the English to German translation task, we selected 8 of them which are presented in Table 2. Among the systems selected, three of them were online statistical MT systems, whose outputs were gathered during the workshop by translating the data using the provided interfaces and therefore they have been anonymized. The outputs of the 8 selected systems that were submitted in the English-German translation task were also filtered, selecting only the segments corresponding to the 500 reference segments nominated earlier. The end result was 9 test files, one for the reference file and one for each of the systems considered.

No.	System	Description
1	DFKI	German Research Center for Artificial Intelligence (Vilar, 2012)
2	JHU	John Hopkins University (Ganitkevitch et al., 2012)
3	KIT	Karlsruhe Institute of Tehnology (Niehues et al., 2012)
4	OnlineA	Online anonymized SMT system
5	OnlineB	Online anonymized SMT system
6	OnlineC	Online anonymized SMT system
7	UK	Charlse University-Zeman (Zeman, 2012)

Table 3: The systems participating in the system level evaluation

After this initial step of filtering the data, we first evaluated the outputs of the 8 systems using the WCDG parser without the predictor component. This was achieved by calculating the HWCM score, with a maximum length of 6 for a chain, for every pair of reference and translation segments corresponding to a system. The final score of the system was obtained by computing the average of all the intermediate scores. We also evaluated the 8 systems using the altered version of WCDG which incorporates the predictor component. The results obtained are presented in Table 3 together with the ranks of the systems according to human judgments.

No.	System name	Predictor OFF	Predictor ON	Human
1	DFKI-Berlin	0.2042	0.2041	8
2	JHU	0.2234	0.2239	7
3	KIT	0.2377	0.2375	4
4	OnlineA	0.2500	0.2507	2
5	OnlineB	0.2508	0.2513	1
6	OnlineC	0.2268	0.2269	5
7	UK	0.2012	0.2014	6
8	Uedin-Williams	0.2383	0.2380	3
ρ score		0.92	0.92	

Table 4: Results of system level evaluation

Evaluation of the metric at system level was performed by measuring the correlation of the HWCM metric with human judgments using Spearman's rank correlation coefficient ρ. In order to compute the

ρ score, the scores attributed to every system by CESM, were converted into ranks. The ρ scores measured were 0.92, for both the implementations of the WCDG parser. This suggests that this method of evaluation might not be enough fine grained in order to capture the differences between the different parses of a sentence.

3.4.3. Segment level evaluation

The first step in evaluating at segment level was filtering the initial test set provided by the NAACL workshop. Similarly to the system level evaluation, 3500 reference and translation segments were selected, that had the length between 50 and 140 characters The test set created was then evaluated using HWCM and the two versions of the WCDG parser. We decided to evaluate using different lengths of the headword chains in order to better capture the difference made by the predictor component. The Kendall tau rank correlation coefficient was calculated in order to measure the correlation with human judgments.

Length	Predictor OFF	Predictor ON
2	0.008	0.012
3	0.009	0.015
4	0.009	0.016
5	0.006	0.019
6	0.016	0.018

Table 5: Results of segment level evaluation

It can be observed that the use of the predictor component has improved the correlation with the human judgments. This proves that parser errors have a negative impact on the performance of the HWCM syntactic metric. The rather low correlation can be explained by the fact that the test set is only a subset of the initial data set. As a reference, in the results reported by the NAACL workshop (Callison-Burch et al, 2012), the best correlation for evaluation of English to German translations is reported to be 0.19, while the worst correlation is reportedly 0.04.

An example of an improved parsing analysis is presented in Table 5. The reference can be translated as "*This is almost from the beginning a moving book.* ", while the translation can be translated as "*This is a moving book, almost from the beginning.* ". The parsing of the translation shows that the parser assigned wrong regents to both "*eine (a)*" and the "*bewegende (moving)*" tokens. However, in the parsing of the reference it can be observed that both "*ein*" and the "*bewegendes*" were analyzed correctly. When the translation is parsed with the predictor turned on, a mapping between the two sentences is made. Based on this mapping, among the predictions made there are ones that state that the regent of the "*eine*" and the "*bewegende*" tokens should be the "*Buch (Book)*" token. When parsing the translation with the predictor turned on, the parser confirms these two predictions and modifies the dependency tree according to them.

Reference: *Dies ist fast von Anfang an ein bewegendes Buch.*
Translation: *Das ist eine bewegende Buch, fast von Anfang an.*

Analysis of the reference:

Dies	→	*is*	*Anfang*	→	*von*
ist	→	*t*	*an*	→	*ist*
fast		*0*	*Buch*	→	*an*

von	→	*ist*	*ein*	→	*Buch*
.	→	*ist*	*bewegendes*	→	*Buch*
	→	*ist*	*.*	→	*0*
		0			
Analysis of the translation predictor turned off:					
Das	→	*ist*	*eine*	→	*bewegende*
ist	→	*0*	*bewegende*	→	*0*
Buch	→	*ist*	*Anfang*	→	*von*
fast	→	*ist*	*an*	→	*von*
von	→	*von*	*.*	→	*0*
,	→	*0*			
Analysis of the translation predictor turned off:					
Das	→	*ist*	*eine*	→	*Buch*
ist	→	*0*	*bewegende*	→	*Buch*
Buch	→	*ist*	*Anfang*	→	*von*
fast	→	*ist*	*an*	→	*von*
von	→	*von*	*.*	→	*0*
,	→	*0*			

Table 6: An example of an improved analysis

4. Dissemination

Following articles were published or accepted to be published:

Melania Duma, Cristina Vertan and Wolfgang Menzel: A new syntactic metric for evaluation of Machine Translation, Proceedings of the ACL SRW 2013 workshop (Acceptance Ratio: 47%)

Mirela Stefania Duma, Cristina Vertan, Integration of Machine Translation in On-line Multilingual Applications - Domain Adaptation, TC3: "Translation: Computation, Corpora, Cognition", Journal Vol 3, No 1 (2013): Special Issue on Language Technologies for a Multilingual Europe

to appear:
Melania Duma, Mirela Stefania Duma, Cristina Vertan, Walther v. Hahn, "Translation Technology for Terminology Translation in Higher Education", Post-conference

Proceedings of the International Symposium "Language for Special Purposes", Vienna, July 2013

The work is now continued through a Ph.D project at the university of Hamburg. We envisage the improvement of the syntactic measure and embedding also semantic information, as well as more detailed testing. Once stable the code fort he evaluation metric will be available as open source

5. Financial Issues

The funding of the project consisted of 4 000 Euro.

The sum was used as follows:

- Research Grant Melania Duma March 2012 – 1000 euro
- Support to Research Grant Melania Duma April-Mai 2012 – 2x 500 Euro = 1000 Euro
- Support to Research Grant Melania Duma Juni 2012 and September-December 2012 – 5x 400 Euro = 1000 Euro
-

References

CALLISON-BURCH C., OSBORNE M. and KOEHN P., *Re-evaluating the Role of Bleu in Machine Translation Research*, Proceedings of EACL-2006, 2006.

FOTH, K: *"Writing weighted constraints for large de-pendency grammars",* Recent Advances in Dependency Grammar, Workshop COLING 2004, 2004.

GIMENEZ, J.: *"Empirical Machine Translation and its Evaluation",* Ph. D thesis, 2008.

KOEHN P. and MONZ C, *Manual and Automatic Evaluation of Machine Translation between European Languages*, NAACL 2006 Workshop on Statistical Machine Translation, 2006.

KÜBLER, S. and PROKIC J.: *"Why is German Dependen-cy Parsing more Reliable than Constituent Parsing*?", Proceedings of the Fifth International Work-shop on Treebanks and Linguistic Theories, 2006.

LIU, D. and GILDEA, D.: *"Syntactic Features for Evaluation of Machine Translation",* Proceedings of ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization, 2005.

MENZEL W. and SCHRÖDER I., *Decision Procedures for Dependency Parsing Using Graded Constraints,* Workshop On Processing Of Dependency-Based Grammars, 1998

NIEßEN S., OCH F. J., LEUSCH G. and NEY H., *An Evaluation Tool for Machine Translation: Fast Evaluation for MT Research,* Proceedings of the 2nd International Conference on Language Resources and Evaluation (LREC), 2000.

OWCZARZAK, K., VAN GENABITH, J., and WAY, A.: "*Dependency-based automatic evaluation for machine translation*", Proceedings of SSST, NAACL-HLT 2007 / AMTA Workshop on Syntax and Structure i*n* Statistical Translation, Rochester, New York, 2007.

PAPINENI K., ROUKOS S., WARD T. and ZHU W.-J., *Bleu: a method for automatic evaluation of machine translation*, RC22176 (Technical Report), IBM T.J. Watson Research Center, 2001.

POPOVIC M., *Class error rates for evaluation of machine translation output*, Proceedings of the Seventh Workshop on Statistical Machine Translation, 2012

PORTER M. F. 1. *An algorithm for suffix stripping. Program*, Vol. 13. Nr. 3, pp 130-137. 1980

SCHRÖDER I., MENZEL W., FOTH K. and SCHULZ M., *Modeling dependency grammar with restricted constraints*, Traitement Automatique des Langues (T.A.L.), 2000

TOUTANOVA K., and KLEIN, D. and MANNING, C., and , and SINGER, Y.. 2003. *Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network*. In Proceedings of HLT-NAACL 2003, pp. 252-259.