A Novel Dependency-Based Evaluation Metric for Machine Translation

Karolina Owczarzak

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

Doctor of Philosophy (Ph.D.)

to the

Dublin City University
School of Computing

Supervisors: Prof. Josef van Genabith and Prof. Andy Way

April 2008

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

Signed: _____ (Candidate) ID No.: _____ Date: _____

‘While I’m still confused and uncertain, it’s on a much higher plane, d’you see, and at least I know I’m bewildered about the really fundamental and important facts of the universe.’
Treatle nodded.
‘I hadn’t looked at it like that,’ he said, ‘But you’re absolutely right. He’s really pushed back the boundaries of ignorance.’ (. . .)

They both savoured the strange warm glow of being much more ignorant than ordinary people, who were only ignorant of ordinary things.

Terry Pratchett, *Equal Rites*

Contents

Abstract

Automatic evaluation measures such as BLEU (Papineni et al. (2002)) and NIST (Doddington (2002)) are indispensable in the development of Machine Translation (MT) systems, because they allow MT developers to conduct frequent, fast, and cost-effective evaluations of their evolving translation models. However, most of the automatic evaluation metrics rely on a comparison of word strings, measuring only the surface similarity of the candidate and reference translations, and will penalize any divergence. In effect, a candidate translation expressing the source meaning accurately and fluently will be given a low score if the lexical and syntactic choices it contains, even though perfectly legitimate, are not present in at least one of the references. Necessarily, this score would differ from a much more favourable human judgment that such a translation would receive.

This thesis presents a method that automatically evaluates the quality of translation based on the labelled dependency structure of the sentence, rather than on its surface form. Dependencies abstract away from the some of the particulars of the surface string realization and provide a more "normalized" representation of (some) syntactic variants of a given sentence. The translation and reference files are analyzed by a treebank-based, probabilistic Lexical-Functional Grammar (LFG) parser (Cahill et al. (2004)) for English, which produces a set of dependency triples for each input. The translation set is compared to the reference set, and the number of matches is calculated, giving the precision, recall, and f-score for that particular translation. The use of WordNet synonyms and partial matching during the evaluation process allows for adequate treatment of lexical variation, while employing a number of best parses helps neutralize the noise introduced during the parsing stage.

The dependency-based method is compared against a number of other popular MT evaluation metrics, including BLEU, NIST, GTM (Turian et al. (2003)), TER (Snover et al. (2006)), and METEOR (Banerjee and Lavie (2005)), in terms of segment- and system-level correlations with human judgments of fluency and adequacy. We also examine whether it shows bias towards statistical MT models.

The comparison of the dependency-based method with other evaluation metrics is then extended to languages other than English: French, German, Spanish, and Japanese, where we apply our method to dependencies generated by Microsoft's NLPWin analyzer (Corston-Oliver and Dolan (1999); Heidorn (2000)) as well as, in the case of the Spanish data, those produced by the treebank-based, probabilistic LFG parser of Chrupała and van Genabith (2006a,b).

Acknowledgements

This work would not have been possible without the help of several people.

First, I'd like to thank Andy for his continuous encouragement to submit conference papers, even when I argued that anything less than a Complete Theory of the Universe is not worth presenting. And for making sure that my judgments, or even judgements, were consistent.

Thanks to Josef, who always had a thousand and eleven research ideas per minute and did not hesitate to share them with us, even if "this might sound completely crazy". In the end, it's often the crazy that works.

Cheers to Grzegorz for getting the Spanish LFG parser ready and for his patient bug-hunting through 16,000 segments of data. Thanks to Aoife and Jennifer for their help with the English LFG parser.

Finally, I want to thank Microsoft Ireland and Enterprise Ireland (grant SC/2003/0282) for funding my work on this thesis. Special thanks go to Microsoft for providing the data for some of the experiments.

Chapter 1

Introduction

Although professional estimate performed remains by person standard still if it walks about process of gauging quality of engine translation, automatic measures have gained big popularity also, mainly as they assure (provide) inexpensive and fast estimate of quality of translated text.

The above sentence was produced by an online translation system PolTran[1], from a Polish counterpart of my original opening line: *While informed human judgment is still the benchmark in assessing Machine Translation (MT) quality, the use of automatic evaluation metrics has become widespread, mainly because such metrics are an inexpensive and fast way to test translation quality.* How good a translation is it? Would you understand it if you read it somewhere? Is it correct English? The first clause in the translation sounds rather cryptic, but the remainder is quite clear, despite some non-standard word order and stacked prepositional phrases. BLEU says the translation is worth zero points and it is completely wrong. NIST votes for 1.6719 out of 5.3981, whatever that means. My friend Sara gives it 3.5 out of 5 and a special award for including alternative translations in parenthesis. My mom says: *I have absolutely no idea what it means* (but that's probably because she doesn't speak English).

[1] http://www.poltran.com/

Evaluating translation quality is almost as complex as the process of translation itself. Even in the case of the "benchmark human judgment", the final verdict on that first sentence may depend on whether the evaluator is acquainted with the subject matter of the text, what their level of education is, whether they have done such tasks before, who is their favourite writer, how they are feeling at the moment, or whether they had their morning coffee yet. To control this variability, researchers devised guidelines and frameworks for successful and reliable assessment, usually involving multiple evaluators and data samples, in order to approximate the ideal "objective" score.

However, with the proliferation of Machine Translation (MT) technologies, there was also a growing need for faster and less labour-intensive methods. Developers of MT systems could not afford to employ a team of evaluators and wait days for the results every time they made a change to the current model. More recently, with the advent of Statistical MT (SMT) and minimum error rate training, which relies on repeated quality testing, employing human evaluation for these purposes became completely impractical. Recent years have witnessed a growing interest in automatic MT evaluation metrics, focusing on the development of new and more accurate ways in which human judgment can be imitated automatically on a wider scale.

Many of the most popular metrics are limited to a simple string match: they compare the surface forms of the candidate translation and a reference sentence (or sentences) in the target language, and, in one way or another, calculate how many elements they share. Sometimes these elements are words, sometimes word sequences such as n-grams. This intuitive simplicity is at the same time their main shortcoming: since the structure of language is hierarchical rather than linear, not to mention the abundance of synonymous expressions, these methods cannot accept alternative yet legitimate forms in which the same concept may be expressed.

It is clear from human translation studies that there is no such thing as *the* correct translation. For instance, take these four parallel human translations used as references

in one of the shared translation tasks, shown in example (1.1).

(1.1) (a) Players are now allowed to take off their sports clothing after scoring as a way of celebration provided that this would not affect the normal process of the game.

(b) FIFA permits football players to take off their uniform to celebrate a score. The prerequisite is that such act will not interfere with the normal proceedings of the game.

(c) Players will be allowed to take off their team shirts in celebration of goals, as long as it does not affect the normal going of the match.

(d) The players are allowed to take off their shirts to celeberate goals, on condition that the normal process of the game should not be disrupted.

Obviously, if there was such a thing as *the* correct translation, the problem of MT itself as well as its evaluation would have been long solved. Instead, we are faced with a great variability of equally valid expressions, leading to difficulties in automatic evaluation. In the case of the simplest metrics based on string comparisons, anything that diverges from the reference, in terms of lexical choices or word order, is marked down. Multiple references as in example (1.1) counteract this effect only partially. Another flaw of the string-based metrics is that they may be inherently biased towards statistical MT systems, as shown in Callison-Burch et al. (2006b), who also confirm that as a result of these deficiencies, these metrics tend to show poor correlation with human judgment on the segment level. We take a closer look at the existing evaluation metrics in Chapter 2 and examine synonyms and paraphrases as a means to improve the evaluation quality in Chapter 3.

To avoid the limitations of string-based metrics, which compare only the surface forms of translated sentences, research has progressed towards examining deeper linguistic levels of translated text in order to assess its quality. This thesis explores one

such area: MT evaluation based on dependency relations[2]. We present a method that automatically evaluates the quality of translation based on the labelled dependency structure of the sentence, rather than on its surface form. Since dependencies abstract away from the particular surface string realization, they provide a more "normalized" representation of (at least some) syntactic variants of a given sentence. Potential lexical variation is accommodated by exploiting synonyms provided by WordNet[3], similarly to Banerjee and Lavie (2005) and Kauchak and Barzilay (2006).

The method presented here substantially extends earlier research of Liu and Gildea (2005), who assess MT quality by calculating n-gram matches on sequences of unlabelled head-modifier dependencies harvested from syntactic trees by head-extraction rules. By contrast, we use labelled dependencies and overlap between flat sets of triple encodings of the translation and reference dependencies. In a direct comparison with Liu and Gildea's (2005) results, we show that our method is better able to reflect human judgment, due to its focus on local grammatical relations rather than tree paths, and the presence of labels defining the type of grammatical relation that connects the head and the modifier, such as *subject*, *determiner*, *adjunct*, etc. Grammatical relation labels add another layer of important linguistic information into the comparison and allow us to account for partial matches, for example when a lexical item finds itself in a correct relation but with an incorrect partner.

We use Lexical-Functional Grammar (LFG) dependencies produced from the translation and reference text by the treebank-based, wide-coverage, probabilistic LFG parser of Cahill et al. (2004), which generates an unordered set of dependency triples for each input. The translation set is compared to the reference set, and the number of matches is calculated, giving the precision, recall, and f-score for each particular translation. To decrease the amount of noise introduced by the process of parsing and extracting de-

[2]Parts of the research presented in this thesis have been published before in the following articles: Owczarzak et al. (2006a,b, 2007a,b,c, 2008).

[3]http://wordnet.princeton.edu/

pendency information, we employ a number of best parses for the translation and the reference segments in order to find the optimum match. The parser and an overview of the grammar underpinning it are given in Chapter 4.

Chapter 5 discusses the dependency-based method in detail, and investigates how it compares with a number of popular evaluation metrics, such as BLEU[4] (Papineni et al. (2002)), NIST[5] (Doddington (2002)), General Text Matcher[6] (GTM; Melamed et al. (2003); Turian et al. (2003)), Translation Error Rate[7] (TER; Snover et al. (2006)), and METEOR[8] (Banerjee and Lavie (2005)). These experiments show not only that the dependency-based method is one of the best-performing evaluation strategies, but it is also less biased towards statistical translation models than most other metrics.

The method's comparison with BLEU and NIST is tested again in Chapter 6, where we revisit an experiment from the TransBooster project (Mellebeek et al. (2006a); Owczarzak et al. (2006b); Mellebeek (2007), among others). TransBooster is a support technology for MT, which enhances translation quality by decomposing and simplifying the input to a translation engine, and recomposing the resulting output after translation. Despite improvement over the baseline translation showing up clearly in manual evaluation, the recorded increases in BLEU and NIST scores were minimal. This time we re-evaluate the data with the dependency-based method, checking whether it is able to discern the (admittedly subtle) improvements introduced by TransBooster.

In Chapter 7 we examine the method's portability to languages other than English. The format of the dependency data in French, Spanish, German, and Japanese, produced by Microsoft's NLPWin linguistic analyzer (Corston-Oliver and Dolan (1999); Heidorn (2000)), is similar to that generated by the LFG parser, and the application of a treebank-based LFG parser (Chrupała and van Genabith (2006a,b)) for the Spanish data set

[4]http://www.nist.gov/speech/tests/mt/scoring/
[5]http://www.nist.gov/speech/tests/mt/scoring/
[6]http://nlp.cs.nyu.edu/GTM/
[7]http://www.cs.umd.edu/∼snover/tercom/
[8]http://www.cs.cmu.edu/∼alavie/METEOR/

allows us to compare the performance of these two parsing strategies directly. Chapter 8 concludes and outlines potential avenues for further research in automatic evaluation.

The use of dependencies in MT evaluation has not yet been widely adopted, but their utility as an accurate method to determine translation quality is starting to be more appreciated in the community. As PolTrans puts it, *There is important step towards inclusion of (participation of) deepest linguistic knowledge for process of estimate of automatic translation also, that is natural and in this domain of language engineering direction of farthest development (evolution) promising.*[9]

[9]Reference: *It is also an important step towards introducing deeper linguistic knowledge into the process of MT evaluation, which is a natural and promising further direction of development for this area of human language technology research.*

Chapter 2

Machine Translation Evaluation

2.1 Human evaluation of translation quality

The discussion about the standards for MT evaluation reaches almost as far back as the MT research itself. An early example of MT evaluation is the study carried out by John B. Carroll and included in the famous ALPAC[1] report of 1966 (Pierce et al. (1966)). The goal of the study was to establish a standard procedure for evaluating translation quality, according to specifications that would be applicable both to human-produced and machine-translated texts. The Russian-to-English translations in Carroll's experiment (three human translations and three machine outputs) were judged on the basis of three features:

- their intelligibility as independent text, in separation from the original,
- their fidelity to the meaning of the original text,
- their reading/rating times.

[1] ALPAC (Automatic Language Processing Advisory Committee) was established in 1964 by the U.S. Government to assess the progress of human language technology and its prospects.

Intelligibility was scored on a 9-point scale with detailed descriptions of the qualities which the translation was supposed to meet at each level. Fidelity, on the other hand, was judged on a 10-point scale, with similarly detailed descriptions, that measured how much more information over what the translation contained could be obtained from reading a source text or a perfect (reference) translation. Each sentence was evaluated by three monolingual English and three bilingual speakers, and one of Carroll's conclusions is that due to the inter-rater variance, at least three or four raters should be employed in evaluation experiments. Particular attention was also paid to the construction of the evaluation sets: as there were six participating translations, six evaluation sets were created, but the sentences in each set were drawn from more than one translation, and placed in random order, to minimize rater bias.

Perhaps the most comprehensive critical review of MT evaluation methods was published in 1979 by Georges van Slype, on behalf of Bureau Marcel van Dijk for the Commission of European Communities (van Slype (1979)). The report made the distinction between two levels of evaluation: *macroevaluation* (or total evaluation), where the acceptability of an MT system is determined and where two different MT systems can be compared, and the level of *microevaluation*, which measures the improvability of a specific system. The evaluation criteria within these two categories were also assessed with respect to their relative effectiveness and cost. In conclusion, van Slype recommended that there be three types of MT evaluation, deployed as necessary:

- an inexpensive, easy-to-use superficial evaluation (using the macroevaluation methods),
- a more elaborate and expensive in-depth evaluation (applied to a system as needed and using the microevaluation methods), and
- a pin-point evaluation (to assess the improvement made on a specific feature of a specific system).

In the first category, which included what we now understand under the term "MT evaluation", i.e. measuring the overall comparative quality of MT systems, the report listed such methods as intelligibility and fidelity rating, reading time, correction rate, and correction time. In contrast to Pierce et al. (1966), the intelligibility and fidelity were to be assessed on a 4-point scale. As to the properties of the test set, van Slype suggested a length of 5,000 - 10,000 words, constructed of continuous sentence groups excerpted from genres suitable for automatic translation, i.e. scientific, economic, or administrative texts. To neutralize the variance among the raters which seems greater when evaluating MT output than when evaluating human translations, he recommended 4-10 judges to assess the intelligibility of MT-produced text, but only 1 or 2 to assess the intelligibility of the original text (for comparison) or the post-edited machine translation. Similarly, he stated that it is enough to have one evaluator to assess the overall fidelity of the machine translation to the original text; he did note, however, that fidelity assessment is not very dependable, as the evaluator needs specialized knowledge in the case of scientific texts, and, moreover, the fidelity judgment will vary from person to person depending on their preconceived notions about the importance of various parts of the original message. This is also why he rated fidelity assessment as "poor" in effectiveness as an MT evaluation method.

A large number of methods described in van Slype's review, plus a great variety of other evaluation techniques, were later included in the FEMTI (A Framework for the Evaluation of Machine Translation) project[2] which took place between 2004 and 2006 at the University of Geneva, one of 'the power houses' of MT evaluation for a number of years. The framework was developed in order to facilitate the creation of evaluation plans for MT developers or other interested parties. It consists of two related taxonomies, one of which describes potential contexts of use for an MT system, whereas the other lists relevant evaluation metrics for that particular context, including aspects such as fidelity,

[2]http://www.issco.unige.ch:8080/cocoon/femti/st-home.html

comprehensibility, style, well-formedness, fault tolerance, adaptability, etc.

Earlier, in 1991, the U.S. Advanced Research Project Agency launched an evaluation scheme to measure their own MT research progress, and within several years they developed an evaluation standard that was summarized in White et al. (1994). This standard comprised three dimensions along which MT was assessed:

- a comprehension test, where readers would answer multiple choice questions about the content in the translated text,
- adequacy, which measured whether the MT output contained the same information as a professionally created (reference) translation,
- fluency, where evaluators were asked (without access to a reference translation) whether the translation sounded like "good English".

The evaluators in all three tasks were monolingual English speakers, and special care was taken to ensure maximal consistency of judgments and to neutralize any influence the order of data might have on the evaluators. For this purpose, the 30 evaluators were presented with 30 distinct evaluation sets that included samples from all tested systems, but no passage appeared in more than one translation, and the order of the passages in the sets varied so that output from every system preceded output from every other system at least in one of the evaluation sets. Moreover, comprehensibility, adequacy, and fluency were judged on different passages to prevent the "halo effect", where the judgment in one of these categories might be influenced by the score of the translation in another category. Similarly, fluency was evaluated without any reference texts so as not to bias the judges by providing them with a single grammatical translation.

Since then, fluency and adequacy, under these or other names (cf. Pierce et al.'s (1966) intelligibility and fidelity), seem to have taken hold as the main features used for assessing MT quality. These two features, each rated on a 5-point scale developed by

FLUENCY How fluent is the translation?	ADEQUACY How much of the meaning is expressed?
5 = Flawless English	5 = All
4 = Good English	4 = Most
3 = Non-native English	3 = Much
2 = Disfluent English	2 = Little
1 = Incomprehensible	1 = None

Table 2.1: Adequacy and fluency scales developed by LDC (2005). Adequacy is assessed with respect to a reference translation.

LDC (2005), constitute the human evaluation scheme employed in most shared MT tasks today, including the NIST Open Machine Translation campaign[3] or the ACL MT shared task[4]. An example of the scales together with their descriptions focused on English as the target language is presented in Table 2.1.

Noticeably, the LDC scales contain fewer levels than the intelligibility and fidelity scales of Pierce et al. (1966), and the LDC levels have much simpler descriptions. However, the fact that the scales contain an odd number of values (as contrasted with, say, the 4-point scale of van Slype (1979)) means that it provides the users with a convenient 'middle value', which many judges can fall back upon when they find dinstinguishing minor differences in translation quality too difficult. This concern was raised at the SMT workshop at HLT-NAACL 2006, and it was suggested that, at least for the purposes of comparing two or more MT systems, a relative scale ordering the translation with respect to each other would be more useful than assigning each of them an abstract score.

The change in evaluation scales might suggest that with the proliferation of new MT systems and evaluation campaigns, the human evaluation design seems to become progressively more simplified as well. For example, at the ACL 2007 MT shared task

[3]http://www.nist.gov/speech/tests/mt/
[4]http://www.statmt.org/wmt07/

(Callison-Burch et al. (2007)) only 40% of test data was evaluated by more than one judge; while in the IWSLT 2006 evaluation[5] (Paul (2006)), only a sample of the test data was selected for human assessment. This simplification process is understandable if we remember how costly and time-consuming human evaluation is; applying all the countermeasures needed to neutralize evaluator bias and variability (multiple raters, separate samples for each aspect of evaluation, etc.) would place full-blown human analysis beyond the reach of contemporary evaluation tasks. This is especially true considering the constantly increasing number of participating MT systems – for instance, the ACL 2007 MT task involved 15 participating systems, and the NIST 2006 open evaluation over 40 systems.

While reducing the complexity of human evaluation schemes due to overwhelming costs makes the evaluation more available, we have to remember that the results obtained from these simplified evaluations are often less reliable. Callison-Burch et al. (2007) note that in their shared task, the observed inter-rater agreement was as low as 0.25 Kappa[6] for judgments of fluency and 0.226 for judgments of adequacy, and intra-rater agreement (i.e. the average consistency of individual evaluators) was at the level of only 0.537 Kappa for fluency and 0.468 for adequacy. Similar low values were discovered in the IWSLT 2006 evaluation, where the inter-rater agreement was only 0.24 for fluency and 0.31 for adequacy.

This means that human judgment, at least as it is employed in most tasks nowadays, might not always be the ultimate, noise-free gold standard. In fact, statistical significance testing should be applied both to human and automatic scores when these scores are used to rank a number of MT systems, as is the case in contemporary evaluation campaigns. This significance testing will most likely produce a 'soft' ranking where some systems will belong to the same cluster if it is impossible to distinguish between them with

[5] http://www.slc.atr.jp/IWSLT2006

[6] According to Landis and Koch (1977), a Kappa value of 0-0.2 indicates slight agreement, 0.21-0.4 shows fair agreement, 0.41-0.6 is moderate agreement, 0.61-0.8 is substantial, and above 0.8 is almost perfect.

a certain level of confidence, based on their scores. Stroppa and Owczarzak (2007) propose a representation for such clusters and describe a method for comparing two such clusterings, an example of which is given in Figure 2.1.

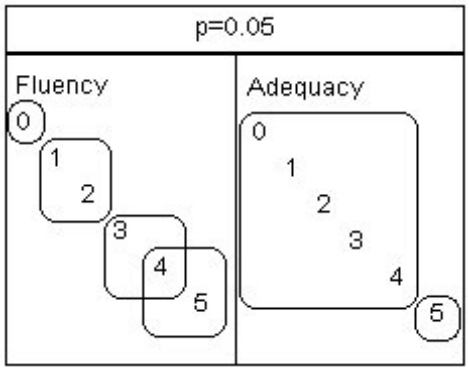

Figure 2.1: Examples of clusterings. Numbers 0-5 represent MT systems; clusters are created on the basis of human fluency and adequacy scores. Relative height of the clusters shows their order in terms of quality. Systems within the same cluster are indistinguishable with a given level of confidence (here 95%).

A single cluster contains systems that are pairwise indistinguishable at a certain level of significance, according to some statistical significance tests such as approximate randomization. By performing this comparison for all pairs of systems, this approach yields an ordered set of clusters. These ordered sets can be then compared with each other. This comparison method is useful when calculating system-level correlations between human and automatic scores (i.e. comparing rankings of MT systems based on human and automatic scores), and will be introduced in Section 5.2.2.

Together with the process of simplifying human evaluation strategies, there has been a noticable growth in the area of automatic MT evaluation metrics. Automatic methods today range in complexity from a simple edit distance (based on Levenshtein (1966)) between the translation and reference strings (Word Error Rate) to support vector machines such as Albrecht and Hwa (2007). The next sections present an overview of some of the most popular and recent examples.

2.2 Automatic MT evaluation

2.2.1 String-based metrics

BLEU

Most contemporary research papers in the realm of MT quote a BLEU score as the measure of their systems' quality. Since its invention, BLEU (BiLingual Evaluation Understudy, Papineni et al. (2002)) has become one of the most popular evaluation metrics, mainly due to its speed, intuitive simplicity, and good correlations with human assessment on the document (or system) level.

BLEU is based on a simple calculation of modified precision. Modified precision counts the number of n-grams in the translation that match the reference (or any of the multiple references) and caps the count by the maximum number of occurrences of a given n-gram in a single reference. In other words, if a translation consists entirely of the word *the* repeated five times, but in one of the references *the* appears only once, and in the other only twice, we are allowed to count only two of the five matching words. This process is applied to any n, but in practice n-grams up to four are used. The modified precision results for the whole document at each n-gram level are combined together using geometric average. Moreover, in order to prevent unfair high precision scores for very short translation sentences, a brevity penalty is calculated over the test set, if the combined length of the translation segments is equal to or shorter than the combined length of best-matching (closest in length) reference segments. The BLEU formula is shown in (2.1); BP indicates the brevity penalty, w are the weights that can be altered for different n-gram levels (but which are uniform in the default setting), and p is the modified precision at a given n-gram level.

(2.1)
$$\text{BLEU} = \text{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right)$$

Note that BLEU was developed with document- or system-level evaluation in mind, and its construction does not allow for high correlation with human judgment on the level of individual segments. At segment level, many sentences such as the one in example (2.2) will be scored as zero for not providing at least one four-gram in common with the reference(s), which artificially lowers their quality score. Segments shorter than four elements, such as the one shown in (2.3), will be scored as zero irrespective of the number of lower n-gram matches.

(2.2) *Translation:* John resigned from his job yesterday.
Reference: Yesterday John quit his job.

(2.3) *Translation:* John resigned.
Reference: John quit.

This systematic underestimation is exacerbated as the number of available references decreases. BLEU's smoothed version uses an add-one technique (where we add 1 to every n-gram count) to counteract this problem, but this, on the other hand, runs the risk of scoring some genuinely bad translations too generously. The problem is that BLEU, like other string-based metrics, evaluates translation quality based on surface comparison of word sequences between a translated sentence and the reference, and is unable to accommodate any legitimate lexical and grammatical variation of the translation, beyond what can be found in the multiple references. For example, the translation shown in (2.4) will fail to obtain an adequate score, because it shares with the reference only two (*John*; *yesterday*) out of three unigrams (*John*; *resigned*; *yesterday*) and none of the higher n-grams.[7] After adding 1 to every count, we have the following number of matches at the individual levels: 1-grams: 3/4, 2-grams: 1/3, 3-grams: 1/2; 4-grams: 1/1; therefore,

[7]In this example we ignore punctuation for the sake of simplicity; however, punctuation is generally included in a typical evaluation.

the geometric average is 0.59 (no brevity penalty, since both sentences are of the same length).

(2.4) *Translation:* John resigned yesterday.

Reference: Yesterday John quit.

While high segment-level correlation with human scores is expected to lead to high document-level correlation, the opposite need not be true. As a result, BLEU, being a metric designed with the goal of document-level reliability, has been widely criticized for its inadequate accuracy of evaluation at the segment level, for example in Callison-Burch et al. (2006b).

NIST

This shortcoming has motivated the creation of a number of other metrics, which have tried to improve on the BLEU baseline correlation with human judgments. For example, BLEU's closest relative NIST (Doddington (2002)) factors in the information score of an n-gram, based on its frequency in the document. Less frequent n-grams are thought to be more crucial to the meaning of a text, and so translating them correctly should count for more than correctly translating frequently used determiners or other function words. Instead of geometric average, arithmetic average is used to combine results from all levels up to 5-grams, and the brevity penalty is adjusted to minimize the impact of small length variations. The full formula for the NIST score is shown in (2.5).

(2.5)

$$\text{Score} = \sum_{n=1}^{N} \left\{ \sum_{\text{matching } w_1 \ldots w_n} \text{Info}(w_1 \ldots w_n) \Big/ \sum_{\text{all trans } w_1 \ldots w_n} (1) \right\} \cdot \exp\left\{ \beta \log^2 \left[min\left(\frac{L_{sys}}{\overline{L}_{ref}}, 1 \right) \right] \right\}$$

Here the precision is modified by the information weights Info($w_1 \dots w_n$), computed on the basis of the frequency of a given n-gram in relation to the frequency of a 'parent' (*n-1*)-gram, in accordance with the equation (2.6).

(2.6)

$$\text{Info}(w_1 \dots w_n) = \log_2 \left(\frac{count\ w_1 \dots w_{n-1}}{count\ w_1 \dots w_n} \right)$$

The brevity penalty in formula (2.5) is calculated depending on the relation of translation length L_{sys} and the average reference length L_{ref}, and β is chosen to make the brevity penalty factor equal to 0.5 when the translation length is 2/3rds of the average reference length.

Note that including information weights (which are particular to a given text) in the scoring process means that the maximum score will differ from document to document. If we score a translation-reference pair consisting only of the sentences in (2.4), NIST gives it 1.0566 points out of the maximum 1.5850 possible (the maximum would be reached if the translation was word-for-word identical to the reference).

General Text Matcher (GTM)

In an attempt to improve on BLEU's inability to account for syntactic differences between a translated segment and its reference, some of the alternative metrics concentrate mainly on lowering the importance of word order, like General Text Matcher (GTM) (Melamed et al. (2003); Turian et al. (2003)). GTM uses the standard notions of precision, recall, and their composite f-measure, to evaluate translation quality, as shown in (2.7)–(2.9), where C indicates candidate translation, R reference translation, and *MMS* is the maximum match size, i.e. the number of matching words capped by the

maximum number of times a given word appears in the reference sentence. The maximum match size prevents double-counting of matching words if the candidate contains more tokens of a given word than the reference.

(2.7)
$$\text{precision}(C|R) = \frac{\text{MMS}(C, R)}{|C|}$$

(2.8)
$$\text{recall}(C|R) = \frac{\text{MMS}(C, R)}{|R|}$$

(2.9)
$$\text{f-measure}(C|R) = \frac{2 \cdot \text{precision}(C|R) \cdot \text{recall}(C|R)}{\text{precision}(C|R) + \text{recall}(C|R)}$$

In the case of multiple references, the maximum matching size is also capped by the mean length of the references. While it also has the option of weighting contiguous sequences of words more than unconnected matching fragments, Turian et al. (2003) conclude from their experiments that such a weight lowers the correlation with human judgment. Turian et al. (2003) also show that GTM outperforms both BLEU and NIST with respect to correlation with human scores, irrespective of the number of references available. Scored by GTM, our simple translation-reference pair in (2.4) would obtain 0.66 points out of 1, which is a straightforward percentage of matching unigrams between the two sentences.

Translation Error (or Edit) Rate (TER)

Another metric, more directly based in edit distance techniques, is Translation Error Rate, also known as Translation Edit Rate (TER; Snover et al. (2006)), which computes the number of substitutions, insertions, deletions, and shifts of words necessary to transform the translation text to match the reference. TER's human-assisted sibling metric HTER, where a human annotator creates a reference that is as close as possible to the system output while retaining all the required information and fluency, has been adapted by the U.S. National Institute of Standards and Technology as the official evaluation metric in their GALE (Global Autonomous Language Exploitation) research program.

Since the TER score is the total number of edits to the candidate translation that would render it identical to the closest reference, divided by the average number of words in a reference, as shown in (2.10), a perfect translation would be scored as 0 and there is no upper bound on the score (i.e. in theory, the translation can be infinitely bad).

(2.10)

$$\text{TER} = \frac{\text{\# of edits}}{\text{average \# of reference words}}$$

Given our translation example (2.4), TER performs one shift (it shifts *John* one word to the right) and two substitutions (it substitutes *resigned* with *yesterday* and *yesterday* with *quit*). As a result, according to the formula in (2.10), 3 edits over 3 words of reference length gives us the final score of 100%.

METEOR

There are several metrics that try to accommodate both syntactic and lexical differences between the candidate translation and the reference, like CDER (Leusch et al. (2006)),

which employs a version of edit distance for word substitution and reordering. Another one is METEOR (Banerjee and Lavie (2005)), which uses stemming and WordNet[8] synonymy. The evaluation in METEOR proceeds in several stages. In the first stage, all exact matches between the translation and the reference words are found. Next, the remaining words are stemmed and the matching process repeats; finally, there is the option of using WordNet to find matches between synonyms among the remaining non-matched words. The final score combines precision and heavily weighted recall at the unigram level with a penalty for non-contiguous matches, as in formula (2.11).

(2.11)

$$\text{METEOR} = (1 - \mathit{penalty}) \cdot \text{f-measure}$$

where

(2.12)

$$\text{f-measure} = \frac{\text{precision} \cdot \text{recall}}{\alpha \cdot \text{precision} + (1 - \alpha) \cdot \text{recall}}$$

(2.13)

$$\text{penalty} = \gamma \cdot frag^{\beta}$$

The penalty is based on fragmentation factor *frag*, which is the number of contiguous matching 'chunks' divided by the number of matching words, with experimentally set values of γ and β. To ensure consistency in this thesis, all experiments where the output translation is in English have been scored using METEOR v0.4.3, with the values

[8] http://wordnet.princeton.edu/

$\alpha = 0.9$, $\gamma = 0.5$, and $\beta = 3$, and only the final experiments involving output languages other than English use METEOR v0.6, with variable settings optimally fit for each input language (Lavie and Agarwal (2007)).

Depending on the choice of modules for METEOR, the example sentence pair (2.4) will obtain different scores. In the "exact" mode, when only the surface form of the words is matched, the translation receives 0.3333 points. Adding stemming does not change the score; however, using stemming and WordNet results in the high score of 0.8519, since *resign* and *quit* are synonyms. This contrast shows clearly how important it is for an MT evaluation metric to be able to account for synonymy. It also shows that synonymy is not the final answer, as the translation-reference pair still fails to obtain a perfect score because of word order variance.

Other approaches

There exist a number of other metrics, which we will only mention briefly here, as we do not use them for comparison in our experiments.

A metric very similar to BLEU, but oriented towards recall rather than precision was proposed by Lin and Och (2004). Recall-Oriented Understudy for Gisting Evaluation (ROUGE) was originally developed to serve as an automatic evaluation metric of machine-generated summaries, but it can also be used to evaluate MT output. It compares the automatic summary/translation to other summaries/translations created by humans by calculating the shared n-gram sequences as well as skip-bigrams, where the match is counted when two words occur in the same relative order, even if they are not immediately adjacent.

TER, discussed in Section 2.2.1, is very closely related to Word Error Rate (WER; Niessen et al. (2000)), which is based on Levenshtein's edit distance (Levenshtein (1966)). The difference between the two metrics is that TER, in contrast to WER, allows shifts of words in addition to substitutions, insertions, and deletions. There are other metrics

related to WER, based on the idea of edit distance. One of them, mWER (Niessen et al. (2000)), allows for multiple references. Position-independent Word Error Rate (PER; Leusch et al. (2003)) treats both translation and reference segments as unordered bag-of-words and computes the number of non-matching elements (i.e. substitutions), adding the difference in word length between the two segments (which is the same as insertion and deletion penalty). At the same time, Leusch et al. (2003) propose Inversion Word Error Rate (invWER), which allows block reordering at a constant cost, i.e. transposition of (bracketed) word sequences from place to place in a sentence.

2.2.2 Dependency-based metrics

The metrics described above use only string-based comparisons, even while taking into consideration differences in word order between the translation and the reference. By contrast, Liu and Gildea (2005) present three evaluation metrics that use syntactic structure and unlabelled dependency information in order to see past the surface phenomena. Two of these metrics are based on matching syntactic subtrees between the translation and the reference, and the third is based on matching headword chains, i.e. sequences of words from a single path in the unlabelled dependency tree of the sentence. Dependency trees are created by extracting a headword for each node of the syntactic tree, according to the rules used by the parser of Collins (1999), where every subtree represents the modifier information for its root headword. The dependency trees for the translation and the reference are converted into flat headword chains, and the number of overlapping n-grams between the translation and the reference chains is calculated. We take a closer look at their dependency chains in Section 5.3, where we compare them to our dependency-based method.

Our method extends this line of research with the use of a term-based encoding of LFG *labelled* dependency graphs into unordered sets of dependency triples, and calculating precision, recall, and f-measure on the sets corresponding to the translation and

reference sentences, as described in detail in Chapter 5. This, with the addition of partial matching and n-best parses, allows us to considerably outperform Liu and Gildea's (2005) highest correlations with human judgment.

The usefulness of relational information (in this case, labelled dependencies) in MT evaluation has been further supported by Callison-Burch et al. (2007). In their presentation of the shared translation task results at the ACL 2007 Statistical MT Workshop, they find that, among all the evaluation metrics they tested, the highest correlations with human judgments for English were obtained by a method based on semantic role overlap developed by Giménez and Màrquez (2007). Giménez and Màrquez (2007) parsed the translation and reference segments into sets of labelled semantic relations from the PropBank Frames (Palmer et al. (2005)), and then calculated how many of these were shared. Intuitively, a method that is able to ignore surface realization of concepts in a sentence is likely to accurately reflect human evaluation of translation content.

Finally, Rajman and Hartley (2001) propose three metrics, C-score, X-score and D-score, which they designed to reflect human judgments of fluency and adequacy. C-score and X-score are used to evaluate the grammaticality of the translated text (with no reference to human-produced translation) on the basis of average spanning parse and average dependency count, respectively. D-score, on the other hand, tests how well the semantic content of a document has been preserved during translation. This is done by computing lexical similarity between the source document and other documents in the source language, and comparing it to the lexical similarity score obtained by the translated document in relation to the translations of these source documents.

2.2.3 Machine learning-based approaches

Recently there have been a number of attempts to apply machine learning methods in the evaluation of MT quality. One such attempt is the linear regression model developed by Russo-Lassner et al. (2005), which makes use of stemming, WordNet synonymy, verb

class synonymy, matching noun phrase heads, and proper name matching as composite features. Kulesza and Shieber (2004), on the other hand, train a Support Vector Machine using features such as proportion of n-gram matches and word error rate to judge a given translation's distance from human-level quality. Another linear regression-based model is presented in Albrecht and Hwa (2007), where the authors additionally show that the method allows for successful use of parallel machine-produced translations of the same text as pseudo-references, which helps overcome the frequent problem of having to procure multiple human-translated references for every test set.

Starting the tradition of naming the evaluation metrics after colours, Akiba et al. (2001) proposed RED (Ranker based on Edit Distances), another edit-distance-based metric for automatic MT evaluation. RED encodes machine-translated sentences with a rank assigned by humans into multi-dimensional vectors from which a classifier of ranks is learned in the form of a decision tree. This decision tree is then used to rank MT output.

The advantage of such approaches is that they make it possible to combine multiple levels of evaluation into a single model, thereby hopefully avoiding the limitations of a single metric. Moreover, the features used in such approaches can be based on linguistic information: for example, Russo-Lassner et al. (2005) use noun phrase head matching in their model, and Albrecht and Hwa (2007) apply a dependency-based language model and headword chains based on Liu and Gildea's (2005) dependency relations. In fact, Ye et al. (2007) find that in their regression model, labelled dependencies were the most useful feature in maximizing the correlation with human judgment, again showing the importance of deep linguistic analysis for MT evaluation. They use a small set of five dependency types (Agent-Verb, Verb-Patient, Modified Noun-Modifier, Modified Verb-Modifier, Preposition-Object), but, significantly, they also find that allowing for partial matching among dependencies increased their utility even further. We show the same effect in Chapter 5 for a much larger set of dependencies.

Whatever the exact design of the evaluation metric, as long as its core process is comparing the translated material to a reference text, its limitations will be similar: inability to account for legitimate variation in lexical and syntactic choices that the translation makes, beyond what is present in the reference. This is a serious impediment, because it is virtually impossible to provide the metrics with *all* potential variants of the reference sentence for each translation. One way to remedy this problem, at least on the lexical level, is to create a collection of synonyms such as WordNet, which the metric can make use of while trying to match the words between the translation and the reference. This will undoubtedly increase the number of matching words, but what about legitimate higher-level phrasal variation? To answer this question, the next chapter discusses automatic generation of synonyms and multi-word paraphrases, which can be used with any evaluation metric to improve its treatment of lexical and low-level syntactic variation, and examines how much improvement we can gain from applying such techniques during evaluation.

Chapter 3

Synonyms and Paraphrases in MT Evaluation

The use of synonyms and paraphrases, such as the WordNet option available in METEOR, is an independent support strategy that can be employed with virtually any evaluation metric to improve its performance. In fact, using synonyms and paraphrases with the existing metrics fulfils the same role as providing multiple reference translations: that is, it increases the likelihood that the alternative translations (in this case created 'on-the-fly' rather than provided with the test set) will contain elements of the MT system output.

Synonym matching plays an important role in Kauchak and Barzilay (2006), who search WordNet synonym sets during BLEU and NIST evaluation to increase the number of matches between the translation and the reference, thereby increasing the correlation with human scores. However, the use of word-level resources such as WordNet helps the existing metrics with only one of their major shortcomings: dealing with permissible lexical variation between the translation and the reference. The other problem, namely, distinguishing legitimate syntactic differences from ungrammatical word order errors, remains unadressed. This is why, in Owczarzak et al. (2006a), we tried to solve both

issues of lexical and syntactic variation through automatic generation of domain-specific multi-word paraphrases rather than externally hand-crafted general synonym resources such as WordNet. We tested a procedure very similar to that of Kauchak and Barzilay (2006), that is, we calculated BLEU and NIST scores allowing for additional matches to occur through the use of multi-word paraphrases derived from the test set through automatic word and phrase alignment. These experiments were motivated by the hope that we can significantly improve existing metrics with simple additional strategies, without venturing into the area of deep linguistic analysis. If this were possible, we would not need complex external tools such as parsers (used in the main method presented in this thesis) or semantic role labellers to accurately evaluate the quality of translation. However, as the next sections show, the increases in correlations with human scores obtained by the automatically generated paraphrases are nowhere near the levels of correlation provided by the labelled dependency-based method.

3.1 Automatic generation of paraphrases

Automatic generation of paraphrases for MT evaluation in Owczarzak et al. (2006a) rests on a combination of two simple ideas. First, two out of three components necessary for automatic MT evaluation metrics like BLEU or NIST – a source language text and a reference text in the target language (the third being, of course, the candidate translation) – constitute a miniature parallel corpus, from which word and phrase alignments can be extracted automatically, much like during the training for a statistical machine translation system. Second, target language words e_{i1}, ..., e_{in} aligned as the likely translations to a source language word f_i are often synonyms or near-synonyms of each other. This also holds for phrases: target language phrases ep_{i1}, ..., ep_{in} aligned with a source language phrase fp_i are often paraphrases of each other. For instance, for the French word *question* the most probable automatically aligned English translations

are *question*, *matter*, and *issue*, which in English are practically synonyms.

Exploiting existing word and phrase alignment techniques from statistical MT for other tasks has been tried with success in several areas. For example, Diab and Resnik (2002) use second language alignments to tag word senses; working on an assumption that separate senses of a L1 word can be distinguished by its different translations in L2, they also note that a set of possible L2 translations for an L1 word may contain many synonyms. Bannard and Callison-Burch (2005), on the other hand, conduct an experiment to show that paraphrases derived from such alignments over large corpora can be semantically correct in more than 60% of the cases. Finally, Callison-Burch et al. (2006a) use the paraphrases derived in this way to improve the quality of machine translation, noting a considerable increase in the coverage of their translation model.

In our experiment, we used a test set consisting of 2,000 sentences, drawn randomly from the test section of the Europarl French–English parallel corpus (Koehn (2005)). The test set was translated by the vanilla phrase-based decoder Pharaoh (Koehn (2004)). The set consisted therefore of three files: the source text in French, the translation in English, and the reference file, also in English. We then used the GIZA++ word alignment software[1] and the refined word alignment strategy of Och and Ney (2003) to produce word and phrase alignments for our miniature bilingual corpus consisting of the source French file and the English reference file.

The generation of paraphrases follows the method of Bannard and Callison-Burch (2005). For each target language word and phrase e_{i1} we collected the source language words or phrases f_{i1}, ..., f_{in} it can be translated as, and then for each of those source language words or phrases f_i we collected its possible translations back to target language e_{i2}, ..., e_{in} (excluding the initial e_{i1}). The target words and phrases were ordered according to the product of the probabilities for each direction of translation, and were placed in a list as paraphrases for e_{i1}.

[1] http://www.fjoch.com/GIZA++

The combination of target–source and source–target lexical mapping produced paraphrases for those target words and phrases that are aligned with more than one source words/phrases or/and whose source language counterpart was aligned with more than one target words/phrases. For example, let us say that the English word *people* is aligned with 4 French words: *personnes* (with probability 0.38), *citoyens* (prob. 0.30), *population* (prob. 0.23), and *signatures* (prob. 0.08). Each of these is in turn aligned with the following English words: *personnes* - *people* (prob. 1); *population* - *population* (prob. 0.62), *people* (prob. 0.37); *citoyens* - *citizens* (prob. 0.77), *people* (prob. 0.18), *public* (prob. 0.04); *signatures* - *people* (prob. 1). When we exclude the initial word *people* from these English translations, and multiply the probabilities (e.g. *people* to *citoyens* 0.30 $*$ *citoyens* to *citizens* 0.77 = 0.231), we obtain a list of potential paraphrases for *people*, ordered according to their likelihood: *citizens*, *population*, *public*. A few more examples are given in (3.1).

(3.1) **area** – field, this area, sector, aspect, this sector

above all – specifically, especially

agreement – accordance

believe that – believe, think that, feel that, think

extensive – widespread, broad, wide

make progress on – can move forward

not true – false

risk management – management of risks

3.2 Domain-specific lexical and syntactic paraphrases

It is important to note here how the paraphrases produced in this way are more appropriate to the task at hand than synonyms derived from a general-purpose thesaurus or WordNet. First, these paraphrases are contextual; they are restricted to only those

relevant to the domain of the text, since they are derived from the text itself. Given the context provided by our evaluation bitext, the word *area* in (3.1) turns out to be only synonymous with *aspect*, and not with *land*, *territory*, *neighbourhood*, *division*, or other synonyms a general-purpose thesaurus or WordNet would give for this entry. To show an example, (3.2) lists the synonyms provided by WordNet for the senses of the noun *area*. Note that *aspect* is not even present in any of the synonym sets, but others like *surface area* or *country* would not be very useful given the domain of the Europarl input text.

(3.2)
- country
- sphere, domain, orbit, field, arena
- region
- expanse, surface area

Our method allows us to produce only such paraphrases as are likely to be useful in the context provided by the source text. Second, the phrase alignment captures something neither a word-level thesaurus nor WordNet will be able to provide: a certain amount of syntactic variation in the multi-word paraphrases. Therefore, we know that a string such as *make progress on* in (3.1), with the underlying part-of-speech sequence verb-noun-preposition, might be paraphrased by *can move forward*, a sequence of modal-verb-adverb.

An advantage of this method is also that the target phrases and words come ordered with respect to their likelihood of being the translation of the source. Each of them is assigned a probability expressing this likelihood, so it is possible to choose only the most likely translations, according to some experimentally established threshold, and so avoid most of the noise resulting from the automatic alignment. The experiment reported here was conducted with a threshold of 0.1, with additional filters that did not permit as paraphrases any numbers, non-word entities, or function words, as well as making sure

that if a multi-word phrase started or ended with a specific function word, its paraphase did too (e.g. ***to*** *rely* ***on*** = ***to*** *depend* ***on***).

3.3 Creating a best-matching reference

After the list of synonyms and paraphrases was extracted from the evaluation bitext, for each reference sentence we created a lattice structure that contained all possible paraphrases for sentence substrings (similarly to Pang et al. (2003)). Then we looked for the maximum scoring path through the lattice, thereby creating a new reference sentence containing only those paraphrases that let us score the corresponding translation higher. Some examples of such references and candidate translations are given in (3.3) – (3.5); the replaced substrings are marked in bold.

(3.3) *Candidate translation:*

the question of climates with is a good example

Original reference:

the climate issue is a good example of this

New reference:

the climate **question** is a good example of this

(3.4) *Candidate translation:*

thank you very much mr commissioner

Original reference:

thank you commissioner

New reference:

thank you **very much** commissioner

(3.5) *Candidate translation:*

i will not add little on matters concerning the substance of the question because we all agree

Original reference:

i will not dwell on the heart of the matter since we are all agreed on that

New reference:

i will not dwell on the **substance** of the **question because** we are all agreed on that

An obvious advantage to generating the alignments from the very test set that is being evaluated is that no other external resources are necessary aside from the alignment scripts. An equally obvious disadvantage is that the quantity and quality of alignments are directly related to the size of the corpus, so a test set of a couple of thousand sentences is unlikely to produce a significant number of paraphrases. In order to examine this effect closer, we compared two cases of paraphrase-enhanced evaluation: one where the paraphrases were generated from the 2,000-sentence test set, and another where they were derived from the complete French–English Europarl training corpus of over 700,000 sentence pairs.

From the 2,000-sentence evaluation bitext we derived paraphrases for 1,524 English words/phrases and placed them in List A. There were 2,188 paraphrases in total, averaging 1.4 paraphrases for each of the 1,524 words/phrases. List B contained paraphrases produced from the full French–English training corpus, and was much larger (on average 1.3 paraphrases for each of the 75,264 words/phrases), while still remaining within the same domain as the test set. In theory, one could generate paraphrases from a number of such parallel corpora (the complete Europarl comes to mind as an ideal candidate) and use them as an external wider-purpose knowledge resource, which would improve on a thesaurus in that it would also include phrase equivalents with some syntactic variation.

First, the translation in our 2,000-sentence set was evaluated by the BLEU and NIST metrics with the original reference; then for each translation sentence a new reference was automatically created, using the paraphrases generated from the source–reference mini-corpus or from the large French–English corpus, such that the reference was the

	BLEU	**NIST**
original ref	0.2131	6.1625
best-match ref (List A)	0.2335	6.6032
best-match ref (List B)	0.2341	6.9494

Table 3.1: Comparison of scores for the original and best-match references for 2,000-sentence test set.

closest possible match for the translation. A subset of 100 sentences was randomly extracted from the test set and evaluated by two independent human judges with respect to adequacy and fluency; the human scores were then compared to the BLEU and NIST scores.

As expected, the use of best-match references produced by our method raised both the BLEU (by .0204 points) and NIST (by .4407 points) scores for the translation of the 2,000-sentence test set produced by Pharaoh, and the increase was greater when more paraphrases were available (additional .0006 and 0.3462 points, respectively). In the case of paraphrases from List A, a new best-matching reference was created in 1,108 cases (55%) of the 2,000 translation–reference pairs. In the case of List B, a new reference was created in 1,640 cases (82%) of the segments. The results are presented in Table 3.1.

The hypothesis that the best-match reference scores reflect better human judgment was also confirmed, as shown in Table 3.2. We asked two human judges to evaluate the 100-sentence subset randomly extracted from our test set with respect to adequacy and fluency, and then we calculated Pearson's correlation between the average human scores and the sentence-level BLEU and NIST scores. Again, the best-matching references created with the use of large-corpus paraphrases (List B) increased the correlations with human scores more than when the paraphrases were generated from the test set itself (List A), but only in the case of BLEU. In the NIST evaluation, the introduction of large-corpus paraphrases actually provided less improvement than List A.

There are two issues that need to be noted at this point. First, at the sentence level, BLEU scored many of the sentences as zero, artificially leveling many of the weaker translations. This explains the low, although still statistically significant[2] ($p < 0.01$) correlation with BLEU for all three types of references. Using a version of BLEU with add-one smoothing we obtain considerably higher correlations. Table 3.2 shows Pearson's correlation coefficient for BLEU, BLEU with add-one smoothing, NIST, and human judgments for the 100-sentence subset. Best-match paraphrase references produced by automatic alignment consistently lead to a higher correlation with human judgment for every metric.

	original ref	best-match ref (List A)	best-match ref (List B)
BLEU	0.297	0.307	0.339
BLEU smoothed	0.397	0.405	0.410
NIST	0.324	0.347	0.332
d	0.498		
d_best	0.506		

Table 3.2: Pearson's correlation between human judgment and BLEU, smoothed BLEU, NIST, using original and best-match references, and between human judgment and the dependency-based method (baseline **d** and best-performing **d_best**).

However, the recorded increase is minimal in comparison to the correlation achieved on the same 100-sentence subset by the baseline and the best-performing versions of our dependency-based method, which is also presented in Table 3.2.[3] We will postpone the detailed discussion of the dependency-based method to Chapter 5; suffice it to say at this point that the baseline version of the dependency-based method only relies on calculations of overlapping labelled dependencies, without recourse to synonyms or paraphrases

[2] A critical value for Pearson's correlation coefficient for the sample size between 90 and 100 is 0.267, with $p < 0.01$.

[3] According to a resampling/bootstrapping test, all the rises in scores are significant; however, according to the Fisher's z' transformation (Fisher (1990)) and the general formula for confidence, the only significant difference exists between the best-performing version of the dependency-based method and the remaining scores.

or other improvements, while the best-performing version uses, among others, WordNet synonyms.

Perhaps generating paraphrases from an even larger corpus and applying more sophisticated filters would make paraphrase-enhanced string-based evaluation more competitive, but for the time being it seems like linguistic analysis in the form of dependency overlap is a more promising candidate for accurate evaluation. One problem here may be that, despite the filters, there is still a significant amount of noise in the automatic alignment/generation process. This is visible in the test we conducted to assess the quality of our paraphrases with respect to their syntactic and semantic accuracy, the results of which are shown in Tables 3.3–3.4. We selected four samples of data from our automatically generated paraphrases; two of these samples were from List A (test set) and two from List B (large corpus). For each list, one of the two samples (sample BEST) contained only one best paraphrase for each entry (as determined by the product of the probabilities), while the other (sample ALL) listed all possible paraphrases. We then evaluated the quality of each paraphrase with respect to its syntactic and semantic accuracy. In terms of syntax, we considered the paraphrase to be accurate if it had the same syntactic category as the original word/phrase; in terms of semantics, if they were semantically equivalent in at least one of their senses.

percentage correct	sample BEST	sample ALL
List A	70%	52%
List B	74%	41%

Table 3.3: Syntactic accuracy of paraphrases.

percentage correct	sample BEST	sample ALL
List A	79%	73%
List B	89%	43%

Table 3.4: Semantic accuracy of paraphrases.

Although it has to be kept in mind that these percentages were taken from relatively small samples (each containing paraphrases for 100 entries), an interesting pattern emerged from comparing the results. It seems that the average syntactic and semantic accuracy of ALL paraphrases is inversely related to corpus size (it decreases as the corpus size increases), but the accuracy of the one BEST paraphrase improves. This reflects the idea behind word alignment: the bigger the corpus, the more potential alignments there are for a given word, but at the same time the better their order in terms of probability and the likelihood to obtain the correct translation. The results of this test were higher than those reported in Bannard and Callison-Burch (2005), who derived their paraphrases automatically from a corpus of over a million German–English Europarl sentences, and achieved the syntactic and semantic accuracy of the best paraphrases (those with the highest probability) of 48.9% and 64.5%, respectively. This is probably because their assessment of paraphrase accuracy was more strict than ours: in Bannard and Callison-Burch (2005), the test was conducted by replacing a phrase with its one most likely paraphrase and checking whether the sentence remained syntactically well-formed and retained its meaning.

In general, it seems that the use of paraphrases is rather unlikely to raise single-handedly the quality of evaluation provided by existing string-based metrics. Synonyms and paraphrases are undoubtedly a valuable addition to MT assessment, but, as we show in Chapter 5 on the example of our dependency-based method, they are only one of the supporting devices that make up an accurate automatic evaluation metric. In the remaining experiments, we use WordNet synonyms instead of multi-word paraphrases explored here for two reasons: first, to allow for a more direct comparison with METEOR that also uses WordNet, and second, because using multi-word paraphrases to create the best-matching reference, as determined by the dependency-based score, would quickly become extremely complex, given that the method involves parsing the input sentences, something that is not straightforward to incorporate into the dynamic reference creation

process. Before we describe the dependency-based method, we examine its theoretical underpinnings, giving an overview of the Lexical-Functional Grammar, and we test the parser on which the method relies in Chapter 4.

Chapter 4

Tools: Lexical-Functional Grammar and the LFG Parser

4.1 An overview of Lexical-Functional Grammar

The dependency-based evaluation method relies on the assumptions of Lexical-Functional Grammar (LFG; Kaplan and Bresnan (1982); Bresnan (2001)). In LFG sentence structure is represented in terms of constituent structure (c-structure) and functional structure (f-structure). C-structure represents the word order of the surface string and the hierarchical organisation of phrases in terms of Context-Free Grammar (CFG) trees. F-structures are recursive attribute-value matrices, representing abstract grammatical relations, such as subject, object, oblique, adjunct, etc., approximating to predicate-argument structure or simple logical forms. C-structure and f-structure are related by means of functional annotations (attribute-value structure equations) present as labels in c-structure trees.

While c-structure is sensitive to surface rearrangement of constituents, f-structure abstracts away from (at least some of) the particulars of surface realization. The sentences *John resigned yesterday* and *Yesterday John resigned* will receive different tree

representations, but identical f-structures, as shown in Figure 4.1.

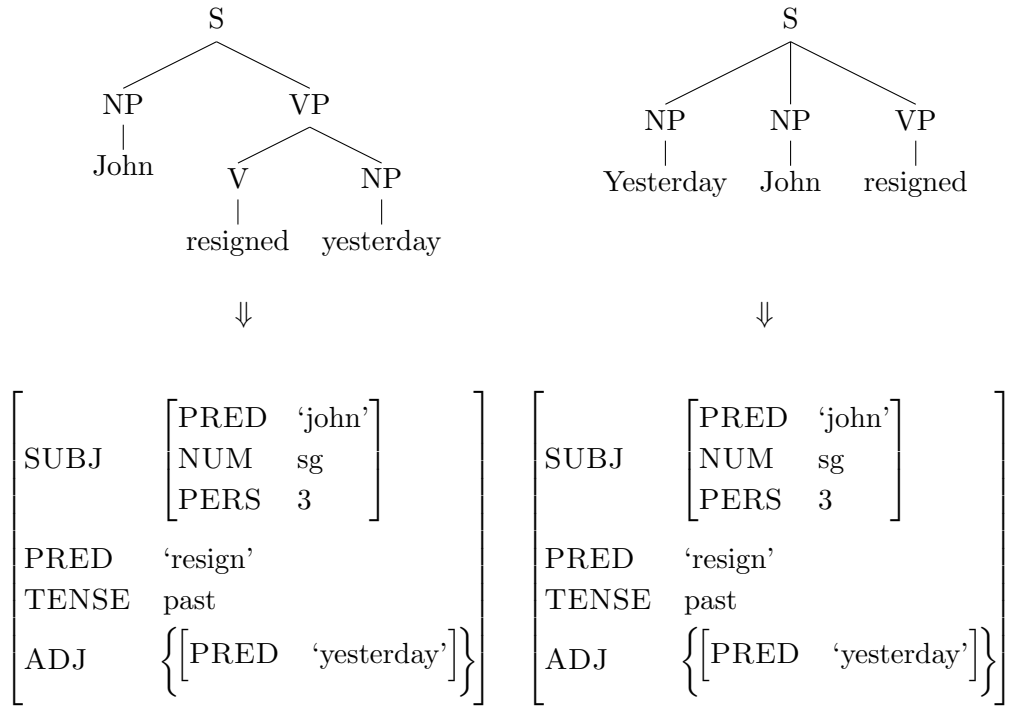

Figure 4.1: C-structure and f-structure

The f-structure in LFG can also be described in terms of a flat set of triples. In triples format, the f-structure from Figure 4.1 (identical for both sentences) would be represented as the set in example (4.1).

(4.1) *subj*(resign,john)
pers(john,3)
num(john,sg)
tense(resign,past)
adj(resign,yesterday)

If these sentences were a translation–reference pair, they would be unlikely to receive

an appropriate score from string-based metrics. For instance, in a calculation similar to the examples in Section 2.2.1, BLEU with add-one smoothing gives this pair a score of 0.76. This is because, although all three unigrams from the 'translation' (*John*; *resigned*; *yesterday*) are present in the reference (*yesterday*; *John*; *resigned*), the 'translation' contains only one bigram (*John resigned*) that matches the 'reference' (*yesterday John*; *John resigned*), and no matching trigrams.

The LFG dependencies abstract away from such common syntactic variations as adjunct placement (cf. the surface position of *yesterday* in both examples) and coordination (e.g. *John and Mary* and *Mary and John* would be given the same representation). Further, our modifications described in detail in Section 5.1, which disregard certain dependency types, also lead to glossing over such differences as topicalization (as in *His job, John was never happy with* vs. *John was never happy with his job*). However, since the dependencies do not offer a fully-fledged semantic analysis, they will provide different representations for phenomena such as active and passive voice utterances (e.g. *John was fired by Mary* vs. *Mary fired John*), or diathesis alternation (e.g. *Mary gave John a book* vs. *Mary gave a book to John*).

Cahill et al. (2004), in their presentation of a set of Penn-II Treebank-based LFG parsing resources, distinguish 32 types of dependencies, including grammatical functions and morphological information. This set can be divided into two major groups: a group of predicate-only dependencies and non-predicate dependencies. Predicate-only dependencies are those whose path ends in a predicate-value pair, describing grammatical relations. For example, for the f-structure in Figure 4.1, predicate-only dependencies would include subject and adjunct: *subj*(resign,john), *adj*(resign,yesterday). Other predicate-only dependencies include:

(4.2)

apposition	*complement*
open complement	*coordination*
determiner	*object*
second object	*oblique*
second oblique	*oblique agent*
possessive	*quantifier*
relative clause	*topic*
relative clause pronoun	

The remaining non-predicate dependencies are:

(4.3)

adjectival degree	*coordination surface form*
focus	*if*
whether	*that*
modal	*number*
verbal particle	*participle*
passive	*person*
pronoun surface form	*tense*
infinitival clause	

These 32 dependencies, produced by the LFG parser of Cahill (2004) described in the next section, and the overlap between the set of dependencies derived from the translation and the reference segments, form the basis of our evaluation method, which we discuss fully in Chapter 5.

4.2 The LFG parser

In the area of parser evaluation, the quality of automatically produced LFG f-structures can be checked against a set of gold standard sentences annotated with f-structures by a linguist. The evaluation is conducted by calculating the precision and recall between the

set of dependencies produced by the parser, and the set of dependencies derived from the human-created f-structure. Usually, two versions of f-score are calculated: one for all the dependencies for a given input, and a separate one for the subset of predicate-only dependencies.

The parser used in this paper and developed by Cahill et al. (2004) obtains high precision and recall rates in such tests.[1] As reported in Cahill et al. (2008), the version of the LFG parser that we use in these experiments (with the LFG annotation algorithm applied to Charniak's (2000) parser output) achieves an f-score of 85.66 in a dependency-based evaluation on the DCU 105 test set, and 86.97 on the Wall Street Journal Section 23 test set; other versions obtain even higher scores. The LFG parser is robust as well, with coverage levels exceeding 99.9%, measured in terms of complete spanning parse.

4.2.1 Parsing MT output

Despite the robustness and high quality of the LFG parser, there might be concern about its ability to deal with MT output, which is often seriously ungrammatical. In other words, the concern is that the parser output for very noisy data is inadequate to support MT evaluation. Applying the usual methods to evaluate parser quality, i.e. comparing automatically produced dependencies against a hand-crafted gold standard, would be less than straightforward in this case. It is difficult to imagine that there can exist one 'gold standard' parse for a heavily ungrammatical segment, or that human annotators would achieve greater than chance agreement on how to parse it, especially given that annotating perfectly well-formed input is not free from ambiguities either. Therefore, we decided to use task-based evaluation to assess the quality of the parses produced from MT text: we test how well our MT evaluation method, based on automatically produced dependencies, correlates with human assessment of the translations, depending on the quality of the translations.

[1] A demo of the parser can be found at http://lfg-demo.computing.dcu.ie/lfgparser.html

Our dependency-based MT evaluation reflects the same process that underlies the evaluation of parser-produced f-structure against a gold standard; we parse the translation and the reference, and then, segment by segment, we check the set of labelled translation dependencies against the set of labelled reference dependencies, counting the number of matches. We collect the number of matches for all segments and then compare it to the total number of dependencies present in the translation and reference texts, calculating precision, recall, and f-score for a given test set. If dealing with highly noisy input decreases the quality of the parser and the annotation algorithm, introducing even more noise than was originally present in the data, we would expect that our dependency-based method will show lower correlation with human judgments for low-quality translations than for good-quality ones.

To test this, we used the data from the Linguistic Data Consortium Multiple Translation Chinese (MTC) Parts 2 and 4, which consists of multiple translations of Chinese newswire text, four human-produced references, and segment-level human scores for a subset of the translations.[2] Although a single translated segment was always evaluated by more than one judge, the judges used a different reference every time, which is why we treated each translation-reference-human score triple as a separate segment. In effect, the experimental set created from this data contained 16,807 segments. After dividing the segments randomly into 20 bins, we ordered the resulting data sets from the lowest to the highest quality (as represented by the mean human scores of fluency, adequacy, or average human score), and we tested the segment-level correlation of our method with the human scores for each of these sets. Figure 4.2 presents how our method's segment-level correlation with human scores of fluency relates to the average fluency score for a given translation set; Figure 4.3 shows the same for scores of adequacy, and Figure 4.4 shows the relation with average human scores.

It is clear that there is no linear relation between the decreasing (or increasing)

[2]LDC Catalog ID: LDC2003T17, LDC2006T04

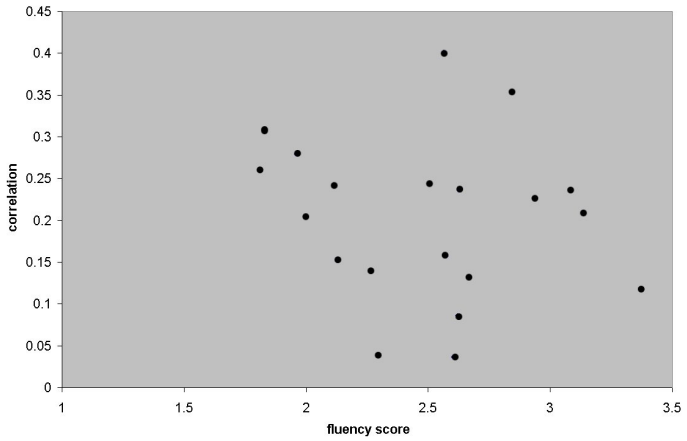

Figure 4.2: Impact of fluency quality on segment-level correlation with human judgment of fluency for dependency-based method

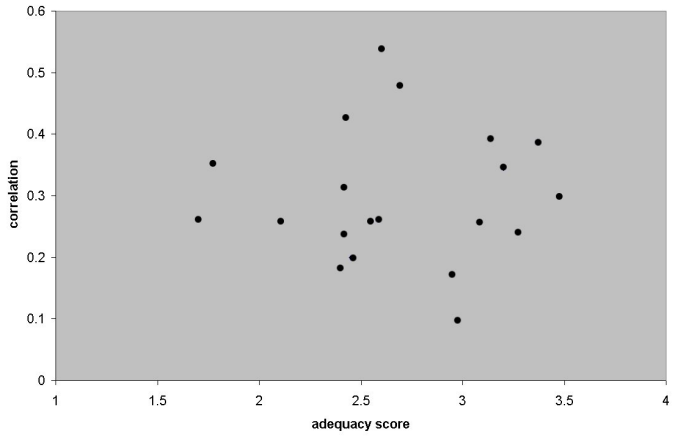

Figure 4.3: Impact of adequacy quality on segment-level correlation with human judgment of adequacy for dependency-based method

quality of input and the LFG parser's ability to produce dependencies from that input. Under each of the three conditions the correlations vary across the 20 translations, as one would expect given the small size of the evaluated samples (around 840 segments each), but the important point is that the variance is not dependent on the translation quality as represented by the fluency, adequacy, or average human score. Neither does

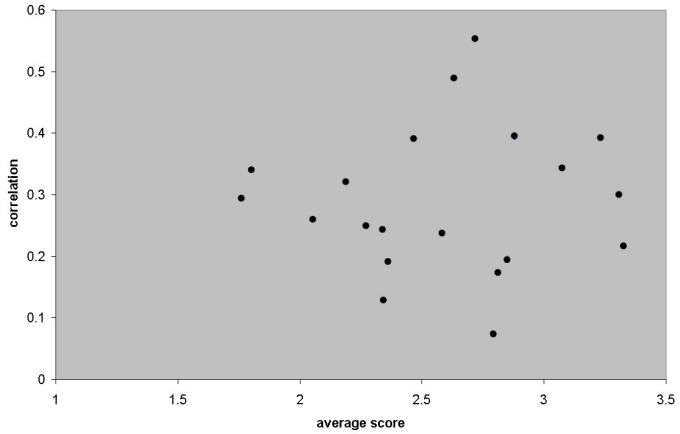

Figure 4.4: Impact of average quality on segment-level correlation with average human judgment for dependency-based method

the quality of translation influence the parser's coverage; we managed to obtain a set of dependencies for each of the segments tested in this experiments.

4.2.2 Dealing with parser noise

Even if the ungrammaticality of input does not seem to prevent the parser from producing dependencies, it is to be expected that certain amounts of noise are nevertheless introduced during the parsing process. This noise would, of course, accumulate when comparing two automatically produced outputs, as is the case in our dependency-based MT evaluation method, in contrast to evaluations of 'pure' parser quality where the comparison is against human-crafted (and therefore, at least in theory, noise-free) dependencies.

To assess the amount of noise that the parser introduces when creating dependencies, in Owczarzak et al. (2007b) we conducted an experiment where a set of 100 English sentences was hand-modified so that the position of adjuncts and coordination order was changed, but the sentence remained grammatical and the meaning was not influenced.

This way, an ideal parser should give both the source and the modified sentence the same f-structure, similarly to the example shown in (4.4).

(4.4) (a) We must change this system, Commissioner.

(b) Commissioner, we must change this system.

The modified sentences were treated like a translation file, and the original sentences played the part of the reference. Each set was run through the parser, and the dependency triples obtained from the 'translation' were compared against the dependency triples for the 'reference', calculating the f-score. Additionally, the same 'translation–reference' set was scored with other metrics (TER, METEOR, BLEU, NIST, and GTM). The results, including the distinction between f-scores for all dependencies and predicate-only dependencies, appear in Table 4.1.

	upper bound	**modified**
TER	0.0	6.417
METEOR	1.0	0.9970
BLEU	1.0	0.8725
NIST	11.5232	11.1704 (96.94%)
GTM	100	99.18
dependency f-score	100	96.56
dependency predicate-only f-score	100	94.13

Table 4.1: Scores for sentences with reordered adjuncts.

The middle column shows the upper bound for a given metric: the score which would be obtained by a perfect translation, word-for-word identical to the reference.[3] The next column lists the scores assigned to the 'translation' containing the reordered adjunct or coordinated elements by each metric we tested. Results show that the dependency and

[3]Two things have to be noted here: (1) in the case of NIST the perfect score differs from text to text, which is why the percentage points are provided along the numerical score; and (2) in the case of TER the lower the score, the better the translation, so the perfect translation will receive 0. Furthermore, there is no upper bound on the score, which makes this particular metric extremely difficult to directly compare with others.

predicate-only dependency scores are lower than the perfect 100, reaching 96.56 and 94.13 points respectively, reflecting the noise introduced by the parser during the parsing and annotation process.

We proposed that the problem of parser noise could be alleviated by introducing a number of best parses into the comparison between the translation and the reference. During the comparison process, for each individual translation-reference segment pair, each of the multiple parses on the translation side is compared against each of the parses on the reference side, and the scores for the best-matching pair are retained. Table 4.2 shows how increasing the number of parses available for dependency comparison brings the process closer to an ideal noise-free parser, resulting in f-scores approximating 99%.

	dependency f-score
1-best	96.56
2-best	97.31
5-best	97.90
10-best	98.31
20-best	98.59
30-best	98.74
50-best	98.79
upper bound	100

Table 4.2: Dependency f-scores for sentences with reordered adjuncts with n-best parses available.

Increasing the number of parses beyond a certain threshold does little to further improve results, and at the same time it considerably decreases the efficiency of the method, so it is important to find the right balance between these two factors. In the experiments presented in the following chapters, we use 50-best parses in our best-performing version of the dependency-based method along with the baseline version that uses the single best parse.

Now that we have presented the parser and the grammar behind it, we can turn to the dependency-based method itself. Chapter 5 compares in detail a number of potential

variations of the method, and tests the method against other popular automatic metrics with respect to segment-level and system-level correlations with human judgment, and examines its potential bias towards certain kinds of translation.

Chapter 5

Dependency-Based Method

As mentioned in Chapter 4, the use of labelled LFG dependencies to evaluate MT output is very similar to the process of parser evaluation. One of the ways to assess the quality of a dependency parser is matching its automatically produced dependencies against human-crafted dependencies created for a given test set, and calculating precision and recall scores. A similar process can be used for MT evaluation: we simply parse the translation and the reference, and then compare the set of dependencies on the translation side with the set of dependencies on the reference side, count the number of matches, and calculate precision, recall, and the f-score. This constitutes the baseline version of our method, with the f-score for a given test set reflecting the quality of translation.

For the experiments presented in this section, we again use the data from the Linguistic Data Consortium Multiple Translation Chinese (MTC) Parts 2 and 4, which consist of 16,807 translation-reference-human score segments. We randomly select 5,007 of these to create a test set, and keep the rest as a training corpus for those versions of the dependency-based method that require the training of weights.

5.1 Versions of the dependency-based method

In this section we examine a number of modifications to the process of dependency matching in order to improve its correlation with human judgment.

Besides the basic version of the dependency f-score, we also looked at the f-score calculated on predicate dependencies only, such as *subject*, *object*, *determiner*, etc., while ignoring 'atomic' features such as *person*, *number*, *tense*, and so on. This focus on predicate-only dependencies turned out not to correlate well with human judgments, leading to the unsurprising conclusion that grammatical features such as *person* and *number* are taken into consideration when human raters are assessing MT quality. This predicate-only version of the method is indicated in Tables 5.3–5.5 as **d_preds**.

Another, more successful addition was the use of 2-, 10-, or 50-best parses of the translation and reference segments, which partially neutralized parser noise and resulted in increased correlations with human scores. Each increase in the number of parses raised the scores further; in Tables 5.3–5.5 we show the versions that used either a single-best parse (labelled as **d** without any number) or those that employed 50-best parses (labelled **d_50**).

In some of the versions we added WordNet synonyms into the matching process to accommodate legitimate lexical variation that might occur between the translation and the reference, and to compare our WordNet-enhanced method with the WordNet-enhanced version of METEOR. These versions are marked in Tables 5.3–5.5 by the addition of the **WN** tag. In each case where WordNet was used, we observe an increase in the correlations.

We also created a version (shown in Tables 5.3–5.5 as **d_pm**) where predicate dependencies of the type *subj*(resign,John) are split into two parts, each time replacing one of the elements participating in the relation with a variable, giving in effect 'split' dependencies such as *subj*(resign,x) and *subj*(y,John). This lets us score partial matches,

where one correct lexical entity happens to find itself in the correct relation, but with an incorrect 'partner'. Interestingly, this turned out to be the best addition to the method; its influence on the baseline version was close to or greater than the use of WordNet (**d_WN**) or 50-best parses (**d_50**). Partial matching can be seen to fulfil a similar role to that of WordNet, namely allowing the match to occur despite the presence of non-identical lexical items. However, it is obvious that their roles are complementary, since their combination in **d_pm_WN** increases the correlations further beyond the levels they achieve when employed separately. It is also possible that partial matching for predicate dependencies contributes to the neutralization of parser noise, i.e. it allows the match to occur even when the dependencies are not completely correct.

In the baseline method, each dependency instance contributes in the same way to the final score for the segment. A single match of *number* between the translation and the reference counts for as much as a single match of *subject* or *object*. This naturally led us to explore the option where some dependency types were weighted more heavily than others. We used the larger part of the MTC corpus (11,800 segments) to train weights for different dependency types in order to maximize the correlation with human scores. The training set was divided into 10 parts, and a 10-fold cross-validation training was performed, using a different part as a local test set each time and the remaining 9 parts as training data. The weights from all 10 training runs were averaged out and used on the previously withheld 5,007-segment section of the data together with other metrics.

The starting point for the training was the case where the final score for a segment consists of partial f-scores for each dependency type occurring in that segment, e.g. the f-score for all *subject* dependencies is added to the f-score for all *number* dependencies, etc., and a mean over these f-scores is calculated. During training, some of these partial f-scores were weighted more than others, such that the correlation with human scores of fluency, adequacy, or with the average human score for these segments was maximized. These versions are shown in Tables 5.3–5.5 as **d_fluency**, **d_adequacy**, **d_average**,

respectively.

We also tried combining weights derived in this way with the proportional weighting implicit in the original method (where, for instance, the (non-)matching of several *number* dependencies contributes more to the final score than a (non-)matching of a single *subject*, by virtue of their count). The combined weights are listed in Tables 5.3–5.5 as **d_ade_prop**, **d_fl_prop**, and **d_av_prop**. These modifications mostly turned out to be a disappointment, as they decreased our correlations on the test set instead of improving them, which suggests that this modification does not carry over well across data sets, or perhaps that more data would be needed for successful training. The only case where the combination of trained weights and implicit proportional weights showed an improvement was in the case of correlations with judgments of fluency (**d_fl_prop**).

The training of weights, however, provided useful information for creating a version of our method in which we exclude from the matching process those dependency types that came out from the training with zero weight (*if*, *modal*, *obl*, *to_inf*, *topic*, *whether*, *xcomp*, *tense*, *that*), which might suggest either that they are not correctly recognized by our parser, that they are very infrequent, or that they carry little relevance for the quality of translation. This modification is indicated in Tables 5.3–5.5 by the **excl** tag in the version label.

Another modification, namely changing the relative rates of precision and recall in the final score, according to values trained on the larger part of the data, did not improve correlations in the test set (**d_excl_pr**). This might have been surprising, given that Banerjee and Lavie (2005) and others report that increased weight for recall consistently improves correlations with human judgments. However, we have to remember that we calculate precision, recall, and f-measure on dependencies (i.e. pairs of words in a grammatical relation) and not on single words, as is the case in Banerjee and Lavie (2005). For metrics operating on the word-level it might be indeed the case that recall gives better prediction of translation quality; if the translation segment contains many

words from the reference (resulting in high recall), it will be seen by human judges as conveying most of the "important concepts", even if not completely fluent. If, on the other hand, most unigrams from the translation segment appear in the reference (giving high precision), it does not necessarily indicate good quality, as this could happen if the translation is short and consists mostly of function words and other less meaningful unigrams. For dependencies the nature of the game changes; we no longer care about words themselves, but about *words as they appear in specific relations with other words.*

A closer examination of the interaction between precision, recall, and human scores for dependencies is shown in Table 5.1. We used the 5,007-segment test set to train new weights for precision and recall in our baseline version of the dependency-based method. Training on the test set, rather than on withheld training data, allows us to see the upper bound on the increases in correlations that could be brought about by changing the weights in this particular test set. Maximizing the correlations with judgments of fluency, adequacy, and average human score separately, we noticed an interesting pattern: while higher weights for recall improve correlations with judgments of fluency, higher weights for precision improve the correlations with adequacy.

d	**fluency**	**adequacy**	**average**
original correlation	0.1529	0.2540	0.2290
max correlation	0.1532	0.2572	0.2300
optimum value	p:0.39 r: 0.61	p: 0.77 r: 0.23	p: 0.66 r: 0.33

Table 5.1: Optimum weights for precision and recall to maximize correlations with human judgments of fluency, adequacy, and average human score. Legend: **d** = dependency-based method (baseline version); **p** = precision; **r** = recall.

This pattern conforms to our intuitions about inherent qualities of translation and reference segments. Human-crafted reference translations are always grammatical, so a translation which contains many of the reference dependencies is bound to be mostly grammatical as well, obtaining high fluency scores. A translation which achieves high precision on its dependencies is likely to contain most "important concepts" from the

reference. Contrary to word-level evaluations, a translation which is a random collection of function and other non-crucial words will not obtain high precision, because the dependencies it generates will be unlikely to appear in the reference.

One could argue that perhaps we could still see the same recall-precision imbalance as in Banerjee and Lavie (2005) in the case of the partial match version of our method. The partial match version, splitting predicate dependencies in two parts, is slightly closer in nature to unigrams, as the element we operate on becomes a single word in a grammatical relation. But, as shown in Table 5.2, retaining the relation label is enough to produce the same patterns as in Table 5.1.

d_pm	**fluency**	**adequacy**	**average**
original correlation	0.1618	0.2677	0.2417
max correlation	0.1623	0.2708	0.2425
optimum value	p: 0.38 r: 0.62	p: 0.72 r: 0.28	p: 0.62 r: 0.38

Table 5.2: Optimum weights for precision and recall to maximize correlations with human judgments of fluency, adequacy, and average human score. Legend: **d_pm** = dependency-based method (partial match version); **p** = precision; **r** = recall.

Given that the increases in correlation we could obtain by changing the precision and recall rates are relatively small, we decided to remain with their harmonious proportion in the standard f-measure, especially since otherwise we would now have to produce three f-scores for each segment instead of one (to correlate with fluency, adequacy, and average judgment).

Finally, we created a version of our method that links the dependencies into flat chains that reflect paths of the tree structure, starting from each leaf node and ending with the root node, as in example (5.1), which shows such chains for the sentence *John should resign tomorrow.*

(5.1) *subj*(resign,john) – *modal*(should,resign)
adj(resign,tomorrow) – *modal*(should,resign)

Then we calculate edit distance between the translation and reference chains that start with the same leaf, counting the maximum edit distance penalty when no such chain was present in the other set. Next we convert the individual edit distance scores into percentages (to normalize across chains of different lengths). A final score for the segment is a combination of scores for all chains. This version enabled a more direct comparison of our labelled dependency-based method with non-labelled headword chains of Liu and Gildea (2005), although they calculated the final score based on n-gram matches rather than edit distance. Surprisingly, this version (labelled in Tables 5.3–5.5 as **d_chain**) caused the biggest drop in our correlation scores. The reason for this is probably that such evaluation puts too much emphasis on the full structure of the whole segment and penalizes any diversion from it. It is likely that this is also the reason for a poor performance of Liu and Gildea's (2005) headword chains in comparison with our method, which will be fully discussed in Section 5.3.

The highest correlation level in all three columns was obtained by a version that employed 50-best parses, partial matching, WordNet, and excluded from the matching process zero-weight dependency types (**d_50_pm_WN_excl**).

In general terms, an increase of 0.03 or more between any two scores in the same column is significant with a 95% confidence interval in Tables 5.3–5.5. The statistical significance of correlation differences was calculated using Fisher's z' transformation and the general formula for confidence interval (Fisher (1990)). According to this calculation, in most cases the difference in performance between versions immediately adjacent in each column in Tables 5.3–5.5 was not statistically significant; however, our best-performing version **d_50_pm_WN_excl** is significantly better than the baseline **d** when it comes to correlations with human scores of adequacy, fluency, and the average human score.

fluency	
d_50_pm_WN_excl	0.1848
d_50_pm_WN	0.1816
d_pm_WN	0.1745
d_50	0.1656
d_fl_prop	0.1640
d_pm	0.1618
d_WN	0.1599
d_excl	0.1544
d	0.1529
d_excl_pr	0.1486
d_fluency	0.1406
d_preds	0.1379
d_chain	0.1310

Table 5.3: Pearson's correlation with human judgments of fluency. Legend: **d** = dependency-based method; **50** = 50-best parses; **WN** = WordNet; **pm** = partial match for predicate dependencies; **fluency** = weights trained to maximize correlation with human judgment of fluency; **fl_prop** = the same weights combined with default proportional weights; **excl** = excluding dependencies with zero weights; **chain** = edit distance on dependency chains; **preds** = predicate-only dependencies; **pr** = smaller weight for precision (0.3) than recall (1.7).

	adequacy
d_50_pm_WN_excl	0.2910
d_50_pm_WN	0.2858
d_pm_WN	0.2759
d_pm	0.2677
d_WN	0.2618
d_50	0.2589
d_excl_pr	0.2576
d_excl	0.2568
d	0.2540
d_ade_prop	0.2236
d_preds	0.2171
d_adequacy	0.2067
d_chain	0.1905

Table 5.4: Pearson's correlation with human judgments of adequacy. Legend: **d** = dependency-based method; **50** = 50-best parses; **WN** = WordNet; **pm** = partial match for predicate dependencies; **adequacy** = weights trained to maximize correlation with human judgment of adequacy; **ade_prop** = the same weights combined with default proportional weights; **excl** = excluding dependencies with zero weights; **chain** = edit distance on dependency chains; **preds** = predicate-only dependencies; **pr** = smaller weight for precision (0.3) than recall (1.7).

	average
d_50_pm_WN_excl	0.2670
d_50_pm_WN	0.2622
d_pm_WN	0.2528
d_pm	0.2417
d_50	0.2381
d_WN	0.2372
d_excl	0.2315
d_excl_pr	0.2293
d	0.2290
d_av_prop	0.2124
d_preds	0.1992
d_average	0.1915
d_chain	0.1796

Table 5.5: Pearson's correlation with average human score. Legend: **d** = dependency-based method; **50** = 50-best parses; **WN** = WordNet; **pm** = partial match for predicate dependencies; **average** = weights trained to maximize correlation with average human score; **ave_prop** = the same weights combined with default proportional weights; **excl** = excluding dependencies with zero weights; **chain** = edit distance on dependency chains; **preds** = predicate-only dependencies; **pr** = smaller weight for precision (0.3) than recall (1.7).

5.2 Comparison with other metrics

Having examined the parser's robustness and quality of output, as well as a number of potential improvements to the dependency-based method, we need to find out whether the method provides a better way to evaluate MT output than existing metrics. For this purpose, we tested the correlation of our method and other metrics with human judgments of translation quality on both system level (i.e. in quality ranking of multiple MT systems) and segment level (i.e. in quality evaluation of individual translated segments). We carried out the segment-level comparison on the same data set as in the experiment in Section 5.1, i.e. 5,007 segments randomly extracted from MTC Parts 2 and 4.

5.2.1 Segment-level correlations

The test set was scored using BLEU, BLEU with add-one smoothing, NIST, GTM, TER, METEOR, and our labelled dependency-based method. Since Section 5.1 discussed and compared a number of versions of the dependency-based method itself, in this section, for clarity, we present the comparison between other metrics and only a few chosen variations of our method: the baseline version, the baseline version with partial matching (which will be useful for comparisons with Chapter 7), and the best-performing version. Table 5.6 shows the correlations of BLEU, BLEU with add-one smoothing, NIST, GTM, TER, METEOR, and the dependency-based method with human judgments of fluency, adequacy, and the average human judgment. As described in Section 5.1, the best-performing version of our method includes the use of the 50-best parses for the translation and reference, partial matching for predicate dependencies, WordNet to account for the remaining lexical variation, and excludes zero-weight dependencies.

Table 5.6 shows that the best version of our dependency-based method is significantly better than any other metric when it comes to correlation with human judgments of fluency (as in Tables 5.3–5.5, any difference of 0.03 or more between scores in the same

fluency		adequacy		average	
dep_best	0.1848	M+WN	0.2913	dep_best	0.2670
dep_pm	0.1618	dep_best	0.2910	M+WN	0.2524
M+WN	0.1536	METEOR	0.2724	dep_pm	0.2417
dep	0.1529	NIST	0.2685	METEOR	0.2367
BLEU_s	0.1452	dep_pm	0.2617	NIST	0.2317
METEOR	0.1451	GTM	0.2599	dep	0.2290
GTM	0.1435	dep	0.2540	GTM	0.2282
TER	0.1420	BLEU_s	0.2054	BLEU_s	0.1955
NIST	0.1396	TER	0.1930	TER	0.1863
BLEU	0.0709	BLEU	0.1270	BLEU	0.1119

Table 5.6: Segment-level correlation with human judgments of fluency, adequacy, and average human score. Legend: **dep** = dependency-based method (baseline version); **dep_pm** = dependency-based method (partial matching); **dep_best** = dependency-based method (best version); **M+WN** = METEOR with WordNet; **BLEU_s** = BLEU with add-one smoothing.

column is significant at the 95% confidence level). In terms of correlations with judgments of adequacy, the method is slightly below METEOR enhanced with WordNet, but the difference is not statistically significant. Our method's advantage over METEOR with WordNet in terms of correlation with average human score is not large enough to be statistically significant either. When it comes to other existing metrics, METEOR occupies the highest position, but the differences between its correlation and the next best candidate is never large enough to be statistically significant. On the other hand, BLEU and TER are lagging significantly behind GTM, NIST and METEOR in terms of correlations with human scores of adequacy and the average human score. Predictably, the smoothed version of BLEU, where we add 1 to every count[1], is perfoming much better at segment-level correlations than the original unsmoothed BLEU, which artificially levels down many weaker translations and scores them as 0. The original BLEU ends up last in all three categories, far behind all competitors, confirming its inappropriateness

[1]If the translation has at least one unigram in common with the reference. If not, the score remains 0, to accurately evaluate sentences which are completely divergent from the reference(s).

as a segment-level evaluation metric.

5.2.2 System-level correlations

To check our method's correlation with human judgments in ranking multiple MT systems, we employed only Part 4 of the MTC data set (also known as TIDES 2003 MT Evaluation data, as used at the ACL 2005 MT evaluation workshop). The data consists of six MT outputs (E09, E11, E12, E14, E15, E22),[2] four human references (E01, E02, E03, E04; of which we use three: E01, E03, and E04, to enable a direct comparison with Liu and Gildea (2005) in Section 5.3), and two segment-level human judgments for each of the translations. For each of the MT systems we averaged out the two human scores on the segment level, and we calculated their average for a system-level score. This was done separately for the scores of fluency, adequacy, and their average. This gave us a ranking of the participating MT systems, with E14 being the best and E22 the worst of the set. We then computed Pearson's correlation between the human system-level scores and system scores assigned by each of the metrics we were comparing. Table 5.7 shows the correlations.

fluency		**adequacy**		**average**	
BLEU	0.746	M+WN	0.961	dep_best	0.908
dep_best	0.662	dep_best	0.955	METEOR	0.903
METEOR	0.651	METEOR	0.952	M+WN	0.893
GTM	0.634	GTM	0.911	GTM	0.867
M+WN	0.602	NIST	0.874	BLEU	0.845
NIST	0.587	BLEU	0.831	NIST	0.825
TER	-0.570	TER	-0.640	TER	-0.649

Table 5.7: System-level correlation with human judgments of fluency, adequacy, and average human score. Legend: **dep_best** = dependency-based method (best version); **M+WN** = METEOR with WordNet.

First, it is important to note that even though there is always a wide disparity

[2]This data set is usually described as containing seven, not six, MT system outputs; however, our release for some reason does not contain the output of system E17.

between segment- and system-level correlations, here this divide is amplified by the fact that due to the structure of the data, all the metrics we tested at the segment level only had access to a single reference, whereas for the system-level evaluations reported here we used three references, which led to overall higher scores for all the metrics. Second, in contrast to its performance on the segment level, BLEU turns out to be considerably better at system-level correlations with human judgments of fluency compared to the other metrics. It also appears to be the most stable metric, in that its correlations with fluency and adequacy are the least divergent, a fact that has been shown in other work as well (cf. Stroppa and Owczarzak (2007)). Other metrics display much higher correlations with adequacy than with fluency, which mirrors the pattern seen in segment-level evaluation.

Of course, because there are so few data points (6 scores, one for each MT system), few of the differences between the correlations are statistically significant. The only significant difference exists between TER and the group containing METEOR with WordNet, the dependency-based method, and METEOR, in the adequacy column. In fact, it is enough to compare this table with the system-level correlations presented in Banerjee and Lavie (2005), who tested on almost the same data (but with the fourth reference E02, as well as MT system E17 which is not present in our release of the data), to see that the ranking of metrics based on system-level correlations is far from stable. Their results for system-level correlations of relevant metrics with the average human judgment are shown in Table 5.8. The F1 measure that they report can be compared to GTM with the exponent set to 1, which is what we used in our tests. In Banerjee and Lavie's (2005) experiments, although the order of METEOR, F1 (≈GTM), and the set containing BLEU and NIST is the same as in Table 5.7, the relative order of BLEU and NIST differs from our evaluation. Additionally, all their correlations with the average human score are slightly higher than those that we report; this could be due to the effect of employing one more reference in the testing process.

	average
METEOR	0.964
F1	0.948
NIST	0.892
BLEU	0.817

Table 5.8: System-level correlation with average human score presented by Banerjee and Lavie (2005)

Since in order to obtain statistically significant correlations on the system level we would have to test a much greater number of systems, we employ instead a cluster comparison method described in Stroppa and Owczarzak (2007). The cluster representation is based on the assumption that when a difference in the evaluation scores obtained by two MT systems is not statistically significant, the significance test places the two systems in the same cluster (i.e. decides that they are indistinguishable with a given level of confidence).

If such significance testing is applied to all the system scores given by each of the metrics discussed here, the output of the evaluation will be a collection of clusterings (where a clustering is a ranking of – possibly overlapping – clusters of systems), instead of the 'hard' rankings (of individual systems). An example of clusterings, given before in Chapter 2, is repeated here in Figure 5.1.

We can then compare the clusterings produced by different metrics, using a modification of the Rand statistics (Halkidi et al. (2001)). For each pair of MT systems, we look at what is their cluster affiliation in clustering A, produced by metric A, and in clustering B, produced by metric B. If the two systems belong to the same cluster in A (i.e. they are indistinguishable according to metric A), and they belong to the same cluster in B as well, then we count this as one instance of agreement between the clusterings A and B. Similarly, if in both A and B System 1 is in a higher-ranking cluster than System 2, we count an instance of agreement. If, however, System 1 belongs to a higher-ranking

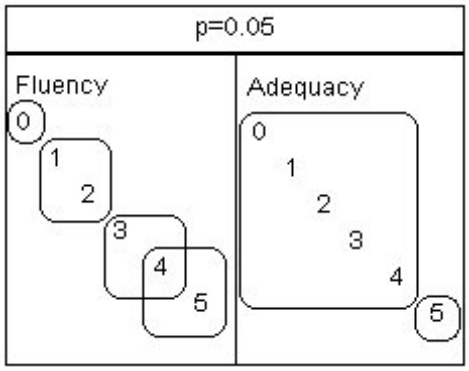

Figure 5.1: Examples of clusterings. Numbers 0-5 represent MT systems; clusters are created on the basis of human fluency and adequacy scores. Relative height of the clusters shows their order in terms of quality. Systems within the same cluster are indistinguishable with a given level of confidence (here 95%).

cluster than System 2 in A, but to a lower-ranking cluster in B, we count this as a strong disagreement between the clusterings. The remaining case is when one of the clusterings is able to distinguish between Systems 1 and 2 and places them in different clusters, but the other one does not. This is then counted as an instance of weak disagreement. Agreements and disagreements between clusterings are scored as follows for each individual comparison of two pairwise rankings $s(a, b)$:

(5.2)

$$s(a,b) = \begin{cases} 1 & \text{if } (a = b) \\ -1 & \text{if } (a =' \ll') \text{ and } (b =' \gg') \\ -1 & \text{if } (b =' \ll') \text{ and } (a =' \gg') \\ 0 & \text{otherwise.} \end{cases}$$

The first case corresponds to an agreement, the second and third cases are strong disagreements, and the last one is a weak disagreement. The calculation of the final score for the comparison of clusterings A and B is shown below.

(5.3)
$$S(A,B) = \frac{2 \times \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} s(A(i,j), B(i,j))}{n \times (n-1)},$$

This formula describes the proportion of agreements to all performed comparisons and yields a value between -1 and 1, to parallel the typical range for Pearson's correlation, where a value of -1 denotes a perfect negative correlation, and a value of 1 denotes a perfect positive correlation.

To test the statistical significance of the system-level scores produced by the metrics under discussion, we apply approximate randomization (Noreen (1989); Riezler and Maxwell (2005)) and produce clusters with the p value = 0.001. We then compare each of the clusterings produced by the automatic metrics (BLEU, NIST, GTM, METEOR, METEOR with WordNet, and the best-performing version of the dependency-based method), with the clusterings generated from applying approximate randomization to the human scores of fluency, adequacy, and the average score for each system. Table 5.9 shows the resulting correlations between the automatic clusterings and those based on human scores.

fluency		**adequacy**		**average**	
M+WN	0.86	M+WN	1	M+WN	0.94
dep_best	0.74	dep_best	0.86	dep_best	0.80
BLEU	0.6	BLEU	0.74	BLEU	0.66
METEOR	0.6	METEOR	0.66	METEOR	0.66
NIST	0.54	NIST	0.66	NIST	0.6
GTM	0.54	GTM	0.66	GTM	0.6
TER	0.46	TER	0.53	TER	0.53

Table 5.9: Correlation of clusterings produced by the automatic metrics and human scores of fluency, adequacy, and the average score. Legend: **dep_best** = dependency-based method (best version); **M+WN** = METEOR with WordNet.

In the cluster comparison, it turns out that after approximate randomization, METEOR enhanced with WordNet places all six MT systems in the same relative cluster

order as the human scores of adequacy, achieving the perfect positive correlation. In this test, it also manages to outperform the dependency-based metric in all three comparisons. Again, all the metrics show the same pattern of being slightly more in tune with the human adequacy scores, reflecting those clusterings better. In constrast to the system-level hard ranking correlations from Table 5.7, GTM and NIST achieve identical level; however, BLEU this time performs as well as or better than METEOR in all three categories. TER remains at the bottom of the scale.

5.3 Comparison with Liu and Gildea (2005)

In the next experiment, our goal was to compare our method directly with the dependency headword chains described in Liu and Gildea (2005), in order to see the impact of using labelled versus non-labelled dependencies for MT evaluation. For this purpose, we tested our method on their dataset (Multiple Translation Chinese Part 4), which is the same data as used in Section 5.2.2. Liu and Gildea (2005) create chains of head-modifier word pairs (with no relation labels) and calculate n-gram matches between the translation and the reference chain sets in a manner very similar to BLEU. For example, in the case of the translation sentence *I have a red pen*, the 2-word headword chain set will include:

(5.4)
(have I)
(have pen)
(pen a)
(pen red)

These chains would be matched against a similar set of 2-word headword chains generated from the reference; a similar matching process would apply at each desired level of chain length. Liu and Gildea (2005) test the correlation of this evaluation with human scores of fluency and overall human judgment. They report correlation results

only for two of the six MT systems, E14 and E15, which have the best and the worst human scores, respectively.[3] They also give the correlation for each n-gram level of their Head-Word Chain Metric (HWCM) metric up to a 4-gram; the highest correlation, however, is reached on the 3-gram level for E14 and on the 1-gram level for E15.

We test our method with the three references that Liu and Gildea (2005) use (E01, E03, and E04), as well as with each of the references separately. We also add to the comparison the **d_chain** version of our method, described in Section 5.1, which is the closest in form to HWCM; we connect our labelled LFG dependencies into chains, and calculate edit distance on the translation and reference chains that start with the same leaf node. Importantly, for the purposes of this experiment, one can draw a parallel between **d_chain** and HWCM for higher level n-grams, but also between our baseline method **d** (which is precision and recall on unordered sets of dependencies) and the 1-gram level HWCM (which is matching individual words).

Table 5.10 presents the results for the high quality translation from system E14 and Table 5.11 presents results for E15, which was judged the worst of all participating systems.

With the exception of one instance (the baseline version **d** with reference 3 for E14), the labelled dependency-based method **d** outperforms HWCM even when it uses only one of the three references that HWCM employs. When all three references are employed, the difference in correlations between HWCM and the labelled dependency-based method is even greater.

What is really interesting is the difference between the original version of our method **d** and the chain version **d_chain**. Note that our chains show higher or comparable correlations with human scores than the original version **d** for E14, the high quality translation, but considerably lower correlations for E15, the low quality translation. We can see the same pattern in HWCM and our method: longer dependency chains are better

[3] According to Liu and Gildea (2005). In our experiments, the system with the lowest average human scores is E22.

	E14	
	fluency	**overall**
HWCM 3 refs	0.130 (1-gram) 0.142 (2-gram) **0.157 (3-gram)** 0.153 (4-gram)	0.191 (1-gram) 0.195 (2-gram) **0.202 (3-gram)** 0.199 (4-gram)
d ref 1	0.203	0.251
d ref 2	0.222	0.270
d ref 3	0.143	0.202
d 3 refs	0.242	0.297
d_chain ref 1	0.247	0.266
d_chain ref 2	0.255	0.267
d_chain ref 3	0.185	0.213
d_chain 3 refs	0.273	0.288

Table 5.10: Pearson's correlation with human judgments of fluency and overall human judgment for HWCM and the dependency-based method on E14.

at reflecting human judgments than unordered dependency sets only for high-quality translations (for E14 HWCM's 3-gram level gave better results than 1-gram; **d_chain** shows higher correlation than **d**), and they are worse for low-quality translation (for E15 HWCM's 1-gram level gave better results than all higher n-gram levels; **d** shows a higher correlation than **d_chain**). The HWCM scores repeated here hint that this effect might be linearly related to the length of the chain, although more data would be necessary to confirm this.

At the same time, it seems that the baseline version **d** and 1-gram HWCM are more dependable across types of input. The correlations with fluency judgments remain stable whether the input is high- or low-quality translation. Correlations with overall human judgment drop somewhat for low-quality translation E15, but the drop is smaller than in the case of **d_chain** or higher n-gram HWCM.

We can conclude from this experiment that, for MT evaluation purposes, it is not advisable to use any method that puts a lot of emphasis on the correctness of long tree-

	E15	
	fluency	**overall**
HWCM 3 refs	**0.128 (1-gram)**	**0.152 (1-gram)**
	0.120 (2-gram)	0.142 (2-gram)
	0.119 (3-gram)	0.144 (3-gram)
	0.113 (4-gram)	0.137 (4-gram)
d ref 1	0.216	0.273
d ref 2	0.181	0.220
d ref 3	0.210	0.256
d 3 refs	0.230	0.275
d_chain ref 1	0.158	0.207
d_chain ref 2	0.126	0.162
d_chain ref 3	0.176	0.215
d_chain 3 refs	0.177	0.217

Table 5.11: Pearson's correlation with human judgments of fluency and overall human judgment for HWCM and the dependency-based method on E15.

derived paths, whether they are picked from a syntactic category tree or a dependency tree. The potential syntactic and lexical variability between the translation and the reference means that the tree structure may vary greatly between the two, and – especially in the case of statistical MT produced from word associations without attention to syntactic well-formedness – even if a probabilistic parser manages to construct a coherent spanning parse from the MT text, such a parse will be most likely very different from a well-formed reference sentence. In such a case the translation might be penalized too heavily, even if it contains some of the correct concepts. The results in Tables 5.10 and 5.11 show that it is better to evaluate the translation segment based on a set of separate short local dependencies, which is the basis of our method; then the proportion of correct and incorrect local relations better reflects human judgment of translation quality.

5.4 Metric bias: SMT vs. rule-based MT

Another issue we want to explore in this chapter is whether our measure is biased towards statistical MT output, a problem that has been observed for n-gram-based metrics like BLEU and NIST. Callison-Burch et al. (2006b) report that BLEU and NIST favour n-gram-based MT models such as Pharaoh (Koehn (2004)), with the effect that translations produced by rule-based systems score lower on the automatic evaluation, even though human judges consistently rate their output higher than Pharaoh's translation. Others have observed this tendency as well; in one of our experiments, reported in Owczarzak et al. (2006a), where the rule-based system Logomedia[4] was compared with Pharaoh, BLEU scored Pharaoh 0.0349 points higher than Logomedia, NIST scored Pharaoh 0.6219 points higher than Logomedia, whereas human judges scored Logomedia output 0.19 points higher (on a 5-point scale) than Pharaoh.

In the experiment reported here we also compared scores for the phrase-based MT system with Pharaoh as the decoder and the rule-based system Logomedia on a data set taken from Europarl (Koehn (2005)). We created a set of 4,000 sentences drawn randomly from the Spanish–English test section of Europarl, and we produced two translations: one by Logomedia, and the other by the standard phrase-based statistical decoder Pharaoh, using alignments produced by GIZA++[5] with the SRILM toolkit[6] and the refined word alignment strategy of Och and Ney (2003). Each translation of the 4,000-segment test set was evaluated with a set of metrics, including our own, and 100-segment samples of the two translations were evaluated by a human judge with respect to fluency and adequacy, using the traditional LDC scale 1-5.

The numbers conformed to the earlier experimental results: for the small manually evaluated subset, Logomedia's translation received an average score of 4.11 for adequacy and 3.97 for fluency, whereas Pharaoh's translation was scored at 3.55 for adequacy

[4] http://www.lec.com/
[5] http://www.fjoch.com/GIZA++
[6] http://www.speech.sri.com/projects/srilm/

and 3.63 for fluency. Clearly, rule-based translation appears to have better quality in the eyes of human judges. However, all the automatic metrics we used to evaluate the 4,000-segment test set gave higher scores to the translation produced by the statistical phrase-based Pharaoh than to the one produced by rule-based Logomedia. This gap in scores is presented in Table 5.12, which shows, for each metric, the percentage by which Pharaoh scores were higher.[7] Note that for TER, which does not have an upper bound, it is not possible to convert the difference in scores to percentages.

metric	**PH score LM score**
TER	1.997
BLEU	7.16%
NIST	6.58%
dep	4.93%
GTM	3.89%
METEOR	3.80%
dep_best	2.80%
METEOR+WordNet	1.56%

Table 5.12: Difference between scores assigned to Pharaoh and Logomedia. Positive numbers show by how much Pharaoh's score was higher than Logomedia's. Legend: **dep** = dependency f-score, **dep_best** = best-performing version of the dependency method (cf. Tables 5.3-5.5).

None of the evaluated automatic metrics were able to reflect Logomedia's advantage established by human assessment. The closest approximation is the score from METEOR enhanced with WordNet, with the best-performing version of the dependency-based metric (also using WordNet) coming in second. It is interesting to speculate why even those metrics that go beyond simple forms of word-string matching (METEOR with its stemming and WordNet synonyms, or our dependency-based metric with the emphasis on local grammatical relations and the use of synonyms) still show a bias (albeit smaller)

[7]The differences between Pharaoh and Logomedia scores were tested for significance using Student's t-test (although perhaps approximate randomization would be a better testing method). According to the t-test, all differences between scores obtained by Logomedia and Pharaoh translations from the tested metrics are significant with $p < 0.005$.

towards the statistical system.

This effect might be particularly pronounced in the case of this experiment, since our test set (the source text and a single reference) is simply a part of the parallel corpus that was excluded from the training of the statistical system. The reason for the bias is then twofold: first, a statistical system has an inherent advantage in being trained on a data set with properties very similar to the test data, even if the exact test data is not a part of the training set. An SMT system learns specific word associations from a corpus, and then reproduces them, in many cases matching word substrings found in the reference that comes from the same corpus. A rule-based MT system has no such opportunity to learn corpus-oriented associations. Second, metrics that rely mainly on matching word sequences between translation and reference intensify this effect. However, even the metrics which do not depend solely on word sequences for evaluation will not lead to a completely fair evaluation in experiments where the test data (or at least the reference) comes from the same corpus as the training data for the participating SMT system. To minimize this effect, tasks involving evaluation of multiple MT systems of different types would need to employ test sets with independently produced references.

5.5 One-sided parsing

Since the dependency-based method requires parsing of both the translation and the reference, and, moreover, its best version employs 50-best parses on each side, it would be reasonable to find ways to improve the method's efficiency and minimize the time and computational power involved to some extent, as long as the improvement comes with no decrease in the method's accuracy. Mehay and Brew (2007) propose a modification of the dependency-based method so that it involves parsing only the reference side, and then checking whether two elements of an unlabelled dependency relation appear in the same surface order in the translation string. This, in their view, will avoid the problem

of incorrect dependencies generated from ungrammatical MT output; however, as we saw in Section 4.2.1, at least in the case of our LFG parser the quality of MT text being parsed does not have any obvious influence on the method's correlation with human judgment.

Mehay and Brew (2007) use the Combinatory Categorial Grammar (CCG) parser of Clark and Curran (2004) to parse the reference file, and strip grammatical labels from the resulting dependencies. For example, a sentence *Please fill in your name* would produce the following dependencies: (*name*, *your*), (*fill*, *name*), (*fill*, *in*), (*please*, *fill*). Then, for every head word, in this case *please*, *fill*, and *name*, they list all dependent words appearing to the left and to the right of that head word in the reference dependencies: for *please* the left context is empty, and the right context contains *fill*; for *fill*, the left context is empty, and the right context contains *in* and *name*; for *name* the left context contains *your*, and the right context is empty. Then they check whether the translation string contains the head word, and if so, whether all the dependent words for that particular head word appear in the left or right string context as appropriate (i.e. in the substring preceding or following the head word, respectively). The percentage of correctly recalled head-dependent pairs is combined with a length penalty in the final score.

Comparing their method, named BLEUÂTRE ('blue-ish'), with the results we published in Owczarzak et al. (2007b), Mehay and Brew (2007) show that BLEUÂTRE falls behind most other metrics; their results are repeated here in Table 5.13. The correlations for all the metrics except BLEUÂTRE are taken from Owczarzak et al. (2007b); the version of our method that Mehay and Brew (2007) choose for direct comparison and include in the table is the predicate-only version (which we abandoned after early experiments because of low correlations with human scores), as it is most similar to the output of their CCG parser. As in Owczarzak et al. (2007b), a difference of 0.015 or more between two metrics in the same column is significant at 95% confidence level.

fluency		**adequacy**		**average**	
BLEU	0.155	METEOR	0.278	METEOR	0.242
OEtAl	0.154	NIST	0.273	NIST	0.238
METEOR	0.149	GTM	0.260	OEtAl	0.236
NIST	0.146	OEtAl	0.224	GTM	0.230
GTM	0.146	BA	0.202	BLEU	0.197
TER	-0.133	BLEU	0.199	BA	0.186
BA	0.128	TER	-0.192	TER	-0.182

Table 5.13: Mehay and Brew's (2007) correlations between various metrics and human judgments of fluency, adequacy, and average human score. **OEtAl** = predicate-only version of the dependency-based method described in Owczarzak et al. (2007a); **BA** = BLEUÂTRE.

To emphasize the efficiency and quality of their method, Mehay and Brew (2007) conduct an experiment where they parse both the translation and the reference, calculating unlabelled and labelled f-score on the dependencies, and then test the correlations of these scores with human judgments. It turns out that the correlations are almost identical to the values of their original method, where they parsed only the reference and used unlabelled dependencies to search for matching two-word sets in the translation strings. This suggests that perhaps we might be able to simplify our method in a similar way without losing quality.

In order to test this hypothesis, we re-implemented Mehay and Brew's (2007) method with our LFG parser, generating dependencies only from the reference file, stripping the grammatical function labels, compiling left and right contexts for each head word, and then matching the head-dependent pairs with the translation string. We then combined the number of matches with a length penalty shown in example (5.5), the same as in Mehay and Brew (2007), and we analyzed the correlations with human scores of fluency, adequacy, and the average human score on the same set of 5,007 MTC segments from our earlier experiments.

(5.5)

$$LP_{candidate,reference} = \begin{cases} 1, \text{if } length(candidate) < length(reference) \\ e^{(1-\frac{len(candidate)}{len(reference)})}, \text{otherwise} \end{cases}$$

Table 5.14 shows how this one-sided version of our dependency-based method compares to some chosen values from Tables 5.3–5.6, namely the best-performing version of the dependency-based method (50-best parses, partial matching, WordNet, and exclusion of zero-weight dependencies), the baseline version of the dependency-based method (f-score on unordered dependencies), the worst-performing version of the dependency-based method (dependency chains), METEOR with WordNet (best-performing of all other metrics), as well as TER and NIST (worst-performing of all other metrics). As in Tables 5.3–5.6, the difference between any two correlations must be 0.03 or greater in order to be significant at the 95% confidence level.

fluency		**adequacy**		**average**	
dep_best	0.1848	M+WN	0.2913	dep_best	0.2670
M+WN	0.1536	dep_best	0.2910	M+WN	0.2524
dep	0.1529	NIST	0.2685	NIST	0.2317
TER	0.1420	dep	0.2540	dep	0.2290
NIST	0.1396	dep_oneside	0.2051	TER	0.1863
dep_chain	0.1310	TER	0.1930	dep_oneside	0.1830
dep_oneside	0.1182	dep_chain	0.1905	dep_chain	0.1796

Table 5.14: Selected correlations for comparison with the dependency-based method using one-sided parsing. Legend: **M+WN** = METEOR with WordNet; **dep** = baseline version of the dependency-based method; **dep_best** = best-performing version of the dependency-based method; **dep_chain** = edit distance on dependency chains; **dep_oneside** = one-sided parsing version of the dependency-based method.

Unfortunately, it appears that the increased efficiency of one-sided parsing comes with a considerable drop in the quality of the method's performance, contrary to the hypothesis of Mehay and Brew (2007). The new version is significantly worse than

either the best-performing version of the method or METEOR with WordNet in all three correlations; more importantly, it is significantly worse than the baseline version of the method from which it was created. In fact, it is on the same level as the worst-performing candidates of all tested metrics.

There are a number of reasons that might be responsible for this poor behaviour in the case of our method. It seems that the omission of grammatical labels leads to the loss of an important layer of additional linguistic information. Additionally, since we saw in Section 4.2.1 that our probabilistic LFG parser is able to produce labelled dependencies even from ungrammatical MT text without visible decrease in correlations with human judgments, leaving out this component of the evaluation is not likely to greatly enhance its performance. Instead, we introduce a potentially greater amount of noise in the matching process: in order to obtain a successful match, the left- or right-context dependants from the reference have to be found *anywhere* in the translation string to the left or right of the head word, irrespectively of the distance between them and other intervening words. This, as can be easily imagined, is likely to lead to a great number of false positives, for example when the dependant is in a different clause than the head word yet still, linearly, in the appropriate left or right context according to the string search (e.g. an object relation (*buy,book*) from *Mary bought a book* will be incorrectly 'found' in the translation *Mary bought a new typewriter for John, who was planning to write a book about his adventures*).

This chapter has shown that the dependency-based method compares favourably to a number of most popular automatic metrics, both in terms of segment-level as well as system-level correlations with human judgment. The method also shows less bias towards statistical models of translation than most other metrics. What is more, it seems that it is to a large extent the LFG theory and the LFG parser in particular that contribute significantly to the method's results, since in Sections 5.3 and 5.5 our method outperformed dependency-based evaluation metrics based on other frameworks.

However, so far all the experiments involved only English as the target language. The next two chapters test the method for other languages, including Spanish, German, French, and Japanese. In Chapter 6, we describe our work on the TransBooster project, an MT-support technology that improves the quality of translations by decomposing complex MT input into shorter and simpler chunks. However, the improvements evident in manual evaluations of the Spanish output translations were not visible in the automatic evaluations performed by BLEU and NIST. We revisit the Transbooster experiment and apply the dependency-based method to its output in order to see whether it is better able to reflect the improvement than the string-based metrics. Chapter 7, on the other hand, explicitly tests our method and a number of other metrics agains human judgment on large data sets from several languages.

Chapter 6

TransBooster: Wrapper Technology for MT

TransBooster is a wrapper technology designed to improve the output of wide-coverage MT systems (Mellebeek et al. (2005a)) by exploiting the fact that rule-based, statistical, and example-based MT systems tend to perform better when translating shorter sentences than longer ones.[1] For example, given a sentence *The chairman, a long-time rival of Bill Gates, likes fast and confidential deals*, the popular online MT system Systran[2] incorrectly translates it to Spanish as *El presidente, rival de largo plazo de Bill Gates, gustos ayuna y los repartos confidenciales.* In the process of translation, the system has wrongly identified *fast* as the main verb (*ayunar* "to fast") and has translated *likes* as a plural noun (*gustos* "tastes"), rendering the whole translation almost unintellegible.

Working with a parse tree of the input sentence, TransBooster decomposes such complex source language structures into shorter, syntactically simpler chunks, sends the chunks to a baseline MT system and recomposes the translated output into target

[1] Most of this chapter describes work done in a joint research project with Bart Mellebeek. My contribution consisted of debugging the decomposing process, work on the dynamic variables, and conducting the EBMT experiment described in Section 6.2.

[2] http://www.systransoft.com/

language sentences. It has already proved successful in experiments with rule-based and statistical MT systems (Mellebeek et al. (2005b, 2006a)), example-based Machine Translation (EBMT) systems (Owczarzak et al. (2006b)), as well as in experiments with multi-engine MT (Mellebeek et al. (2006b)).

However, those experiments also showed an interesting phenomenon: even though TransBooster did outperform the baseline translation, the differences measured by BLEU and NIST were much smaller than the difference established by manual evaluation of the data. This is not surprising, given that string-based metrics are insensitive to legitimate variance of expression, instead treating all divergence from reference text as equally wrong. As a result, the improvements introduced by TransBooster in grammaticality and lexical choice of the translation go largely unnoticed if they are not present in the reference. Considering that TransBooster was evaluated on test sets with a single reference only, the scope for potential matches was quite narrow.

In this chapter, we revisit an experiment on TransBooster and Example-Based Machine Translation (EBMT) from Owczarzak et al. (2006b), where this effect was most visible, and we re-evaluate the data with our dependency-based method in order to see whether, contrary to string-based BLEU and NIST, it is able to better appreciate the improvements in the grammaticality and lexical content of the translation.

6.1 Simpler input, better translation

As noted before, TransBooster relies on the observation that most existing MT systems give better quality translations to short and simple sentences than to long, grammatically complex ones. Working as an intermediary between the user and an MT system, TransBooster's task is to decompose the input text into short and syntactically simple chunks, send them off to the MT system for translation, and then recompose the translated chunks into a complete translation of the input sentence. The following sections

detail this process.

6.1.1 Decomposing the source sentence

First, using a modified head-lexicalised grammar annotation scheme of Magerman (1995) we find the *pivot* in the syntactic structure of the input sentence. The pivot is generally the main predicate, sometimes with additional lexical elements, such as a particle in the case of a phrasal verb (e.g. *give up*), or a sequence of modal verbs and negation (e.g. *shouldn't have been visible*). This expansion of a pivot beyond the main predicate is required, as the particles and modals are a necessary context for correct translation. Then, we locate all *satellites*, i.e. arguments and adjuncts in the remaining string, applying an adapted version of Hockenmaier's algorithm for CCG (Hockenmaier (2003)). For instance, we deconstruct our example sentence *The chairman, a long-time rival of Bill Gates, likes fast and confidential deals* as follows:

(6.1) [$_{\text{ARG1}}$ The chairman, a long-time rival of Bill Gates,] [$_{\text{pivot}}$ likes] [$_{\text{ARG2}}$ fast and confidential deals].

After we have localized the pivot and the satellites (i.e. the arguments and adjuncts), we create a simpler version of the sentence by omitting all the adjuncts (since they are treated as optional elements) and replacing all the arguments in the sentence with their simpler counterparts, which we refer to as *substitution variables*.

6.1.2 Substitution variables

Substitution variables are short phrases of the same syntactic type as the original we replace (i.e. we replace an NP with an NP), and their purpose is: (i) to help reduce the complexity of the original arguments, leading to an improved translation of the pivot; (ii) to help keep track of the location of the arguments in target. The variables we use can be either static or dynamic, and in choosing the optimal type there is a clear trade-off

between accuracy and retrievability. Static substitution variables are previously defined words or phrases (e.g. the NP *cars* to replace the NP *fast and confidential deals*) that are relatively easy to track in the target, since in most cases we can know their translation by a specific MT engine in advance, but they might distort the translation of the pivot because of syntactic/semantic differences with the original constituent. Dynamic substitution variables comprise the real head of the original constituent (e.g. *deals* to replace the NP *fast and confidential deals*) and guarantee a maximum of similarity, but are more difficult to track in the target.

To track the static variables in target, we use a pre-compiled list of the substitution variables and their most likely translations from a given MT system; in order to track the dynamic substitution variables, we send the variables off to the MT engine as separate segments to be translated, and then localize the output translations in the translations of our simplified sentences. Of course, there is always the risk that the same phrase placed in a particular context will have a different translation than when translated on its own; this likelihood seems lower in the case of the static substitution variables. Our algorithm employs dynamic substitution variables first and backs off to static substitution variables if problems occur.

The result of dynamic substitution in example (6.1) can be seen in (6.2). As a back-off, we also produce the static-variable version in (6.3).

(6.2) [$_{\text{DVAR1}}$ The chairman] [$_{\text{pivot}}$ likes] [$_{\text{DVAR2}}$ deals].

(6.3) [$_{\text{SVAR1}}$ The boy] [$_{\text{pivot}}$ likes] [$_{\text{SVAR2}}$ cars].

These simpler versions of the original sentence are placed on a list of segments to be translated, together with the dynamic substitution variables *the chairman* and *deals* we have just created. Afterwards, we retrieve the translations of the dynamic variables (in this case *el presidente* and *repartos*, respectively), and with a simple string matching

technique we attempt to find them in the translation of the simplified sentence, as in (6.4):

(6.4) [El presidente] tiene gusto de [repartos].

If this string search fails, we back off to the static version. In the static version, we store the translation of the variables beforehand for any given MT system. The risk of mismatch between the 'out-of-context' translation and 'in-context' translation is smaller, because the static variables are intentionally designed to be as simple and unambiguous as possible. If this also fails, the last resort in this process is a back-off to the translation of the whole original sentence.

6.1.3 Satellites

After determining the pivot and constructing the simpler versions of the original sentence, we turn our attention to the satellites: the arguments and adjuncts. If a satellite is shorter than a predefined word count threshold, which varies depending on the syntactic category of the input, we consider it ready to be translated. However, translating sentence fragments out of context is likely to produce a certain amount of noise, so we insert it into a *template* — a simple context that mimics some of the original context from which the satellite was extracted. As in the case of the substitution variables, this context can either be static (a previously established template, the translation of which is known in advance) or dynamic (a simpler version of the original context). The dynamic context for ARG2 *fast and confidential deals* in (6.1) would be a simplified version of ARG1 *the chairman* followed by the pivot *likes*: *The chairman likes.* The translation of the dynamic template is determined at runtime, and then we localize it and remove it from the translated string, taking the remainder *repartos rápidos y confidenciales* as the translation of ARG2 *fast and confidential deals*:

(6.5) [The chairman likes] fast and confidential deals. → [El presidente tiene gusto de] repartos rápidos y confidenciales.

As with the substitution variables, we provide a static context template as well. An example of a static context with a direct object position for simple NPs would be the string *The man sees*, which most of the time in Spanish would be translated as *El hombre ve*, as in (6.6):

(6.6) [The man sees] fast and confidential deals. → [El hombre ve] repartos rápidos y confidenciales.

Since the remaining chunk *The chairman, a long-time rival of Bill Gates* contains more words than a previously set threshold, it is judged too complex for direct translation such as was available to ARG2 *fast and confidential deals*. The decomposition and translation procedure is now recursively applied to this chunk; it is decomposed into smaller chunks, which may or may not be suited for direct translation, and so forth.

6.1.4 Recomposing the translation

As explained in the previous subsection, the input decomposition procedure is recursively applied to each constituent until the input phrases reach a certain threshold, related to the number of lexical items, the syntactic environment of the constituent and the baseline MT system used. Constituents below this threshold are provided with substitution variables or embedded in templates and sent to the baseline MT system for translation. After all constituents have been decomposed and translated, they are recombined to yield the target string output to the user.

For example (6.1), the entire decomposition and recombination leads to an improvement in translation quality compared to the original output by Systran, as is shown in (6.7):

(6.7) *Source:* The chairman, a long-time rival of Bill Gates, likes fast and confidential deals.

Original translation: El presidente, rival de largo plazo de Bill Gates, gustos ayuna y los repartos confidenciales.

Improved translation: El presidente, un rival de largo plazo de Bill Gates, tiene gusto de repartos rápidos y confidenciales.

6.2 Invisible improvement: the EBMT experiment

Interestingly, although TransBooster achieved an improvement over the baseline MT systems in experiments with rule-based, statistical, and example-based MT, the increase in scores it received from automatic scoring metrics like BLEU and NIST was minimal in comparison to what was evident in manual examination of the data. This disparity was most visible in the experiments with EBMT, where, in addition to the usual assessment with BLEU and NIST we also conducted a manual evaluation on a sample of translated data.

6.2.1 Example-Based Machine Translation

In this experiment, we used a hybrid EBMT system developed by Groves and Way (2005), which combines marker-based preprocessing of a parallel corpus with a phrase-based decoder. The alignment of subsentential strings in the source-target corpus is based on the "Marker Hypothesis" (Green (1979)), a universal psycholinguistic constraint which posits that languages are 'marked' for syntactic structure at surface level by a closed set of specific lexemes and morphemes. Given a set of such markers for the source and target languages, their relative position in the sentence, as well as a word-level lexicon, source side chunks are then aligned with their most likely translations in the target sentence. The EBMT system was trained on 958K English-Spanish sentence

pairs from the English-Spanish section of the Europarl corpus (Koehn (2005)).

When translating new input, the process is identical to phrase-based decoding: the system first searches the source side of the corpus to find n-gram matches for the input sentence and subsentential strings, and then retrieves their translations from the target side, combining them according to a language model. The decoding in this experiment was performed using the Pharaoh phrase-based decoder (Koehn (2004)).

6.2.2 Original results

For testing purposes two sets of data were used, each consisting of 800 English sentences. The first set was randomly extracted from Section 23 of the WSJ section of the Penn II Treebank[3]; the second set consists of randomly extracted sentences from the test section of the Europarl corpus, which had been parsed with Bikel's parser (Bikel (2002)).

We decided to use two different sets of test data instead of one because we were faced with two 'out-of-domain' phenomena that have an influence on the scores, one affecting the TransBooster algorithm, the other the EBMT system. On the one hand, the TransBooster decomposition algorithm performs better on gold standard parse-annotated sentences from the Penn II Treebank than on the output produced by a statistical parser such as Bikel (2002), which introduces a certain amount of noise. On the other hand, the EBMT model was trained on data from the Europarl corpus, so it performs much better on translating Europarl data than out-of-domain Wall Street Journal text.

We evaluated the baseline EBMT translation and the TransBooster-enhanced translation of the two test sets using the automatic metrics BLEU and NIST, and the results are shown in Tables 6.1 and 6.2. The evaluation was conducted after removing punctuation from the reference and translated texts, and, in the case of Europarl test set, after removing 59 sentences containing hyphenated compounds that have been incorrectly parsed by the Bikel parser, in effect introducing sentence-level errors in TransBooster

[3]http://www.cis.upenn.edu/∼treebank/

processing. The differences in scores have been tested for statistical significance using the bootstrapping test, and were shown to be significant with 95% confidence level.

Europarl	**BLEU**	**NIST**
EBMT	0.2111	5.9243
TransBooster	0.2134	5.9342
Percent of baseline	101%	100.2%

Table 6.1: Results for EBMT vs TransBooster on 741-sentence test set from Europarl.

PIIT	**BLEU**	**NIST**
EBMT	0.1098	4.9081
TransBooster	0.1140	4.9321
Percent of baseline	103.8%	100.5%

Table 6.2: Results for EBMT vs TransBooster on 800-sentence test set from Penn II Treebank.

For both test sets, the use of TransBooster with the EBMT system outperformed the unenhanced baseline EBMT translation. However, even though the significance of the score differences was confirmed in bootstrapping tests, the recorded increases were very small. Our suspicion was that the scale of improvement in translation quality might not be completely reflected by n-gram measures such as BLEU and NIST, especially as the comparison was carried out against a single reference translation in both cases.

In order to assess the real extent of the introduced changes, we conducted a manual evaluation, where we randomly extracted 100 sentences from the Europarl test set, and compared their baseline translation with that assisted by TransBooster. This evaluation of translation quality was conducted by three native Spanish speakers fluent in English. The judges were given the source English sentence along with the two translations and they evaluated the two translations with respect to fluency and adequacy. In contrast to the generally used evaluation techniques, we used a relative scoring scale instead of the absolute one, i.e. the judges decided which of the two translation (if any) was better

in terms of adequacy and which (if any) was better in terms of fluency.[4] Table 6.3 presents the average percentage of cases when the judges decided that one translation was better than the other in terms of fluency or adequacy. The row "TB > EBMT" indicates the instances when TransBooster output was judged as better than the baseline EBMT output, and the row "EBMT > TB" shows how many times the EBMT output was better than TransBooster output. We also calculated the Kappa coefficient (Fleiss (1971)) which indicates agreement between the judges: the Kappa value in the case of judgments of adequacy was .9259, and in the case of fluency .9178, both of which indicate very high agreement levels.

	fluency	**adequacy**
TB > EBMT	35.33%	35%
EBMT > TB	16%	19.33%

Table 6.3: Manual evaluation for EBMT vs TransBooster on 100-sentence test set from Europarl.

It is clear that TransBooster improved the baseline translation in a fair number of cases, and even if we subtract the percentage of translations where it produced worse output than the unenhanced baseline, we still achieve a 19.33% net improvement over the baseline with respect to fluency, and 15.67% net improvement when it comes to adequacy. These numbers are decidedly higher than the increases of 0.2%–3.8% recorded by the automatic metrics.

6.3 TransBooster re-evaluated

In order to see whether these human judgments would be better reflected by the dependency-based method than by the string-based metrics, we parsed the Europarl data set with the

[4]This relative scale was decided upon following the discussion at the SMT workshop at HLT-NAACL 2006, where the participants suggested that the relative scores would be more useful to comparing two or more MT systems, since with the typical absolute scale (1 to 5) the judges tend to choose the 'safe' middle value of 3, neglecting smaller but still important differences between translations.

treebank-based Spanish LFG parser of Chrupała and van Genabith (2006a,b). During the parsing process, the c-structure is generated by the parser of Bikel (2002) adapted to Spanish, and the f-structure is created by the function labeller and LFG annotation algorithm. The parser also contains modules for morphological tagging and lemmatization (Chrupała (2006)).

The dependencies produced by the Spanish parser are labelled using a similar feature set to the English parser, making adjustments for typological differences between the two languages. The feature set is presented in (6.8).

(6.8)

adjunct	*adjunct relative*	*adjective type*
case	*complement*	*conjunction*
conjunction form	*coordination*	*focus*
form	*gender*	*impersonal*
marker	*mood*	*negation*
number	*object*	*second object*
oblique	*oblique agent*	*passive*
perfect	*person*	*polite*
possessive number	*possessive person*	*predicate link*
pronoun form	*pronoun type*	*particle form*
reflexive	*specifier*	*specifier form*
specifier type	*subject*	*subordination*
subordination form	*tense*	*topic*
verb form	*external complement*	

The Europarl data set was evaluated with two versions of the dependency-based method: the original f-measure calculation over shared dependencies, and another version with partial matching. Unfortunately, we were not able to obtain the Spanish version of WordNet for these experiments; moreover, since the tag set produced by the parser was different from our earlier experiments, we could not be confident without fur-

ther experiments that excluding certain dependency types would improve the method, either. Also, due to parser configuration we could not use n-best parses. Table 6.4 shows the scores obtained by the EBMT baseline translation and the TransBooster-enhanced one.

Europarl	**d**	**d_pm**
EBMT	0.3298	0.3554
TransBooster	0.3284	0.3538
Percent of baseline	99.6%	99.5%

Table 6.4: Dependency-based scores for EBMT vs TransBooster on 741-sentence test set from Europarl. Legend: **d** = dependency-based method; **d_pm** = dependency-based method with partial matching.

Much to our disappointment, the dependencies seem even worse at showing TransBooster's advantage established in human judgment tests. The scores are lower for TransBooster output than for the EBMT baseline. One could argue that perhaps the 100-sentence subset evaluated by human judges was an exception and does not reflect the quality of the whole set; however, even when the evaluation is limited to the 100-sentence subset, still none of the metrics are able to show much difference between the two translations, as can be seen in Table 6.5. In fact, BLEU and NIST both show higher scores for TransBooster than either version of the dependency-based metric.

100-sent	**BLEU**	**NIST**	**d**	**d_pm**
EBMT	0.1689	4.8175	0.3233	0.3470
TransBooster	0.1776	4.8829	0.3236	0.3448
Percent of baseline	105.1%	101.3%	100.1%	99.4%

Table 6.5: Automatic scores for EBMT vs TransBooster on 100-sentence subset. Legend: **d** = dependency-based method; **d_pm** = dependency-based method with partial matching.

Still, a potential criticism of this document-level comparison is that it tries to compare *relative* human judgments with *absolute* automatic scores, which might lead to

skewed results in cases where there is a large variance in the automatic scores for a given test set. Therefore, as a last resort test, we converted all the automatic segment-level scores to relative ranks for the TransBooster and EBMT translations, mirroring the scale used by human judges. Then, using the Kappa coefficient (Cohen (1960)) for pairwise comparison, we calculated the agreement for each of the automatic metrics and each of the human judges, as well as for each pair of the human judges. Then we compared the average inter-judge agreement to the average agreement between a metric and the human scores. In the case of human scores, the average pairwise inter-rater agreement was very high: 0.913 for fluency judgments and 0.948 for adequacy judgments. The Kappa values describing average agreement between human scores and automatic metrics are presented in Table 6.6. Because BLEU is not fitted for segment-level scoring, we also evaluated the data with BLEU using add-one smoothing.

100-sent	**BLEU**	**BLEU_sm**	**NIST**	**d**	**d_pm**
fluency	-0.165	-0.232	-0.110	-0.044	-0.035
adequacy	-0.097	-0.108	-0.076	0.009	0.009

Table 6.6: Average agreement between automatic metrics and human rankings of fluency and adequacy. Legend: **BLEU_sm** = BLEU with add-one smoothing; **d** = dependency-based method; **d_pm** = dependency-based method with partial matching.

Clearly, the automatic metrics are as far away from agreeing with human evaluations as possible. Most Kappa values in Table 6.6 hover around 0, i.e., they show no agreement with judgments of fluency and adequacy at all. The smoothed version of BLEU shows a very weak negative agreement; however, this might be an accidental side-effect of the small sample size which amplifies the influence of individual data points on the final result.

Since neither the string-based metrics nor the dependency-based method (albeit not in its optimal version) seem to be able to indicate the level of improvement generated by

TransBooster, perhaps it is a particularly difficult case for automatic metrics in general. Given such a small data set size, as well as only relative human judgments, we cannot, therefore, conclude anything about the performance of the method in Spanish. In the next chapter, however, we test the method and the Spanish LFG dependency parser on a large set of 15,919 segments, and compare it against the logical form dependencies collected from the Microsoft's NLPWin parse of the same text.

Chapter 7

Dependency-Based Method for Other Languages

Chapter 5 shows that the dependency-based method exhibits considerable potential as an accurate MT evaluation metric, achieving relatively high levels of correlation with human judgments in comparison with other popular metrics on data where English is the target language. Would it be as useful for other target languages? Potential critics of the method might rightly point out that any metric reliant on extraneous resources such as an LFG parser is limited to languages where such resources exist. For instance, we were able to apply our method to Spanish output in Chapter 6, only because the Spanish LFG parser of Chrupała and van Genabith (2006a,b) was available to produce the dependencies in the target language. However, the method can in theory be applied to any types of dependencies and is not limited to LFG parser output. In Section 5.5 we presented research by Mehay and Brew (2007), who proposed an evaluation on unlabelled dependencies produced by one-sided parsing of a reference text with a CCG parser. Still, the results obtained placed the CCG dependency-based evaluation behind most other metrics, suggesting that perhaps not all dependency parsing is equally suited for MT evaluation purposes.

In this chapter we apply the dependency-based method to the output generated by NLPWin, Microsoft's natural language processing tool (Corston-Oliver and Dolan (1999); Heidorn (2000)). Using a knowledge base, grammar rules, and probabilistic information, NLPWin parses input text and produces syntactic structures and logical forms such as the one shown in Figure 7.1.

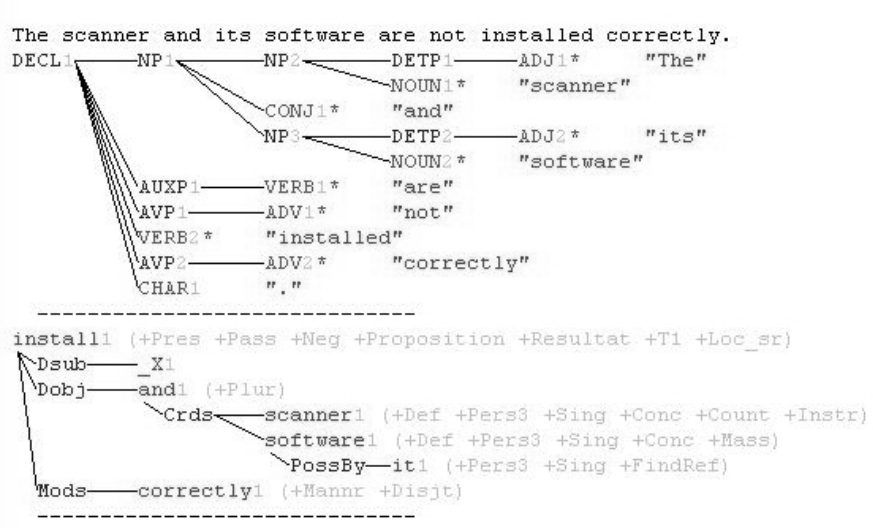

Figure 7.1: An example syntactic parse and logical form produced by NLPWin

NLPWin represents not only categorial information of the input elements, but also their lemma form, grammatical attributes such as person, number, and tense, as well as the functional structure of the sentence, labelling words as *subject*, *object*, *modifier*, etc. Given the example translation–reference pair *John resigned yesterday–Yesterday John quit*, NLPWin produces the syntactic and logical forms in Figures 7.2 and 7.3.

In essence, the NLPWin representation encodes very similar concepts as the LFG parser output, so its adaptation to the dependency-based method is fairly straightforward. NLPWin is available for parsing a number of languages; for the experiments reported here, we examine the logical forms produced from data in French, German, Spanish, and Japanese.

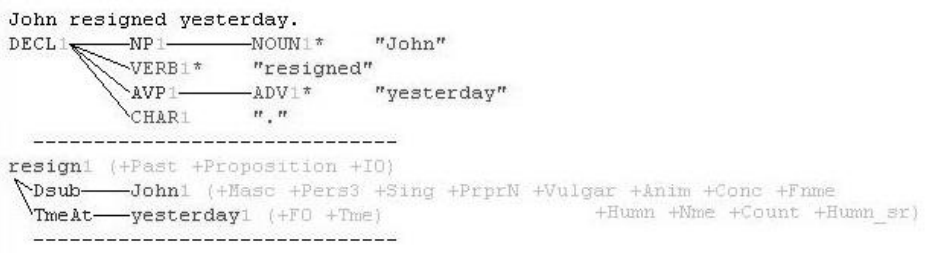

Figure 7.2: A syntactic parse and logical form for *John resigned yesterday.*

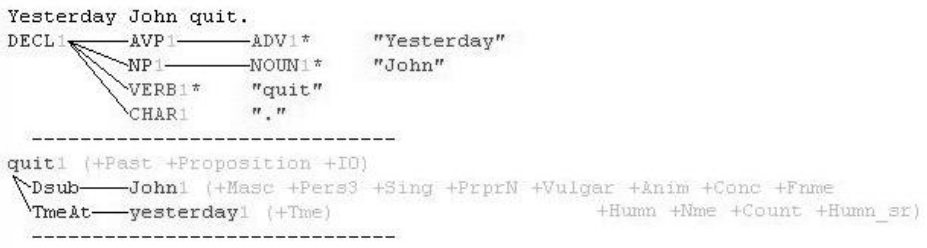

Figure 7.3: A syntactic parse and logical form for *Yesterday John quit.*

The data, generously provided by Microsoft Ireland, consists of English to target language translations of online product support articles, generated by the Microsoft data-driven MT system MSR-MT (Dolan et al. (2002); Richardson (2004)). Typical sample English source sentences from this domain are shown in example (7.1). The total number of segments differs from language to language. The French section of the data consists of 5,834 segments, the German section has 8,109 segments, Spanish — 15,919 segments, and Japanese — 6,845 segments.

(7.1) Not enough memory is available.

The toolbars and menus may change to those of another program, and the X may be replaced with the actual picture.

The Setup.ins file is identical for both the Intel and Alpha platforms.

The product key entered has already been used for a subscription renewal.

Type your new password into the Type your password box, and then click the right-arrow button to log on to the computer.

Each translated segment is accompanied by a single human-produced reference, logical forms of the translation and the reference sentences generated by NLPWin, and corresponding average human scores for the translation. Each translated sentence in French, German, Spanish, and Japanese was judged by 3-5 independent raters on a scale from 1 to 4, described in (7.2), in terms of its similarity to the reference human translation. The scale reflected the overall quality of the translation, without separate scores for adequacy and fluency.

(7.2) 4 = Ideal: grammatically correct, all information included

3 = Acceptable: not perfect, but definitely comprehensible, and with accurate transfer of all important information

2 = Possibly acceptable: may be interpretable given context/time, some information transferred accurately

1 = Unacceptable: absolutely not comprehensible and/or little or no information transferred accurately

We converted the logical forms of the translation and reference sentences so that they mirrored the format of the LFG parser output, resulting in unordered sets of triples such as in (7.3), which is a representation of the example logical form in Figure 7.1.

(7.3)

feature(install,+Pres)	*feature*(scanner,+Conc)
feature(install,+Pass)	*feature*(scanner,+Count)
feature(install,+Neg)	*feature*(scanner,+Instr)
feature(install,+Proposition)	*feature*(software,+Def)
feature(install,+Resultat)	*feature*(software,+Pers3)
feature(install,+T1)	*feature*(software,+Sing)
feature(install,+Loc_sr)	*feature*(software,+Conc)
Dsub(install,_X1)	*feature*(software,+Mass)
Dobj(install,and)	*PossBy*(software,it)
feature(and,+Plur)	*feature*(it,+Pers3)
Crds(and,scanner)	*feature*(it,+Sing)
Crds(and,software)	*feature*(it,+FindRef)
feature(scanner,+Def)	*Mods*(install,correctly)
feature(scanner,+Pers3)	*feature*(correctly,+Mannr)
feature(scanner,+Sing)	*feature*(correctly,+Disjt)

Next, we applied the dependency-based method to the translation and reference triples in order to obtain a score for the translation. However, in these experiments it was impossible to apply the best-performing version of the method for a number of reasons. First, Microsoft NLPWin is essentially a rule-based parser, and while it can in theory produce multiple parses to account for the ambiguities in the text, it will not generate n-best outputs. This means, as with the Spanish experiments from Chapter 6, that we were not able to use the 50-best parses, but were limited to the single output provided with the data. Second, because the set of dependency labels derived from NLPWin logical form differs from that of the LFG parser, we could not exclude zero-weight dependency types. Third, considering that the target language in these experiments was not English, and the use of WordNets for each of the languages (even if they are available) would involve additional complications, we decided to omit the

WordNet-enhanced version of the metric (a limitation that applied also to METEOR).

In effect, the evaluation was conducted using two versions of the dependency-based method: the baseline version (simple f-score calculation on non-modified triples), and the version which used partial matching (cf. Section 5.1), where the triples were 'split' in half and in each case one of the participating elements was replaced by a variable, leading to an increased number of matches when a word occurred in a correct relation, but with a different 'partner' than what the reference contained.

The string representations of the translation and reference segments were then scored with the same set of evaluation metrics as we used in the previous chapters: BLEU, NIST, GTM, TER, and METEOR. Even though METEOR was the only metric explicitly adapted to languages other than English (the languages for which METEOR's "exact" and "stem" options are available include German, French, and Spanish), the remaining metrics can be thought of as language-independent, since all they do is match string objects. The only problematic case here was the Japanese data; even though one could consider a Japanese sentence as a string of symbols, similar to Western languages, Japanese text is not usually segmented into words. Therefore, in order to produce data more compatible with the string-based metrics, we pre-processed the Japanese translation and reference segments with the morphological analyzer JUMAN[1]. However, the output of the analyzer still does not correspond to words in English; rather, it consists of sequences of roots, inflectional morphemes, and derivational morphemes divided by spaces. Additionally, we 'adapted' the string-based metrics to Japanese input in a very basic way by disabling their text pre-processing modules where applicable.

[1] http://www-nagao.kuee.kyoto-u.ac.jp/nl-resource/juman-e.html

7.1 French

The French section of the data consists of 5,834 translation–reference segments with associated average human judgments. The translations were scored with all the metrics used in this experiment, and Pearson's correlation between the segment-level scores and human judgments was calculated. BLEU score was calculated in two ways: using the original formula, where many sentences score zero on the segment level for lack of at least one four-gram in common with the reference, and using add-one smoothing. We also applied two versions of METEOR: one where only exact matches are counted, and another where stemming is performed after the exact match count and the remaining matches are added to the count. Table 7.1 presents the correlations with human judgments achieved by each of the participating metrics in decreasing order. As in previous experiments, the polarity of the correlation does not matter; therefore, TER's scores are treated in absolute terms. The symbols $*$ and † indicate statistical significance of the difference in correlations; each score marked by a symbol is not statistically different (at the 95% confidence level) from the neighbouring scores marked by the same symbol.[2]

First, it is important to notice that all the correlations, in this and following sections, are considerably higher than other comparable (i.e. segment-level) correlations in our earlier experiments. The most probable reason for this is that the human judgment here is the average of 3-5 individual human scores, which normalizes to a certain extent the inherent variability in subjective human evaluation. In contrast, the segment-level evaluations in Chapter 5, while calculated over a test set of a similar size (5,007 segments), only included a single human judgment for each translation. On the other hand, the higher correlations might be due to the fact that the translation domain in the current experiment is very narrow and specialized; consequently, there might be much less potential for vocabulary variance between the translation and the reference than in

[2]As in Chapter 5, the statistical significance of correlation differences in these experiments was calculated using Fisher's z' transformation and the general formula for confidence interval (Fisher (1990)).

French	
METEOR_st	0.647
METEOR_e	0.633
BLEU_s	0.572
NIST	0.545∗
GTM	0.542∗
d_pm	0.491†
TER	-0.485†
d	0.479†
BLEU	0.426

Table 7.1: Correlations with average human judgment for French. Legend: **METEOR_st** = METEOR with stemming; **METEOR_e** = METEOR in "exact" mode; **BLEU_s** = BLEU with add-one smoothing; **d** = dependency-based method; **d_pm** = dependency-based method with partial matching; ∗ and † indicate scores that are not significantly different from each other.

MultiTrans newswire text from Chapter 5. As a result, string-based metrics might be particularly effective on this data set. One additional reason for across-the-board higher correlations is that in these experiments we are correlating a single automatic score designating the overall translation quality with a human score that is also explicitly a single score evaluating the overall translation quality, rather than with a mean of two scores for adequacy or fluency, or adequacy and fluency separately.

While METEOR in both its incarnations obtains significantly higher correlations than any other metric for the French data set, BLEU with add-one smoothing comes close behind, outperforming the remaining competitors. In contrast to its basic, unsmoothed version that shows the least correlation with human judgments, this is quite a good result when it comes to segment-level evaluation, which, as we remember from Chapter 5, was not BLEU's strongest suit. Since add-one smoothing basically adds 'free' points to every translation segment, provided that it shares at least one unigram with the reference, this suggests that the regular version of BLEU consistently underestimates translation quality. In order to examine this issue more closely, we devised a version of smoothed

BLEU that added 1 to the matched and total n-gram counts only when the count for the particular n-gram level was 0 (again, as long as there was at least one unigram in common). Table 7.2 shows that this version showed a markedly lower correlation with human scores on the same data, although still higher than the baseline original BLEU. This means that BLEU's tendency to underestimate translation quality cannot be blamed solely on the geometric average it uses in the calculation, where 0 on any n-gram level results in 0 as the final score. To confirm this finding, Table 7.2 shows another instance of smoothed BLEU, where smoothing was applied across-the-board to every count (provided the sentence had at least one unigram in common with the reference), but the value added was only 10^{-3}, amounting to much less 'free' weight than in other smoothed versions. This smoothing technique correlated better with human scores than the baseline, but worse than the two versions which added 1 to every count. The same effect was present for the remaining target languages.

	French	German	Spanish	Japanese
BLEU_s	0.572	0.525	0.595	0.559
BLEU_s when 0	0.531	0.473	0.583	0.544
BLEU_s low weight	0.495	0.431	0.532	0.478
BLEU	0.426	0.365	0.453	0.429

Table 7.2: Correlations with human judgment for versions of BLEU. Legend: **BLEU_s** = BLEU with add-one smoothing across-the-board; **BLEU_s when 0** = add-one smoothing applied only when the count is 0; **BLEU_s low weight** = add 10^{-3} across-the-board.

Unfortunately, the dependency-based method does not perform very well for this data set. One reason for this might be that, as mentioned before, the limited vocabulary variance of the domain might give an advantage to string-based metrics. Second, it might be that the dependencies derived from NLPWin's logical forms are not as well fitted to the task of MT evaluation as the LFG dependencies, similarly to the CCG output of Mehay and Brew (2007). Both the original and the partial-match versions

are not significantly different than TER, and together they form the worst-performing group of all the metrics except the baseline BLEU.

7.2 German

The German data consists of 8,109 segments and, as in the previous section, it was evaluated with BLEU, BLEU with add-one smoothing, NIST, GTM, TER, METEOR in exact and stemming modes, as well as the dependency-based method with and without partial matching. The results of the evaluations are presented in Table 7.3.

German	
METEOR_st	0.558∗
METEOR_e	0.548∗
d_pm	0.532†
d, BLEU_s	0.525†
NIST	0.505
GTM	0.485
BLEU	0.365
TER	-0.294

Table 7.3: Correlations with average human judgment for German. Legend: **METEOR_st** = METEOR with stemming; **METEOR_e** = METEOR in "exact" mode; **BLEU_s** = BLEU with add-one smoothing; **d** = dependency-based method; **d_pm** = dependency-based method with partial matching; ∗ and † indicate scores that are not significantly different from each other.

Again, METEOR comes out as the winner, although this time with less advantage over the competitors than in the French data set. BLEU with add-one smoothing correlates decidedly better than its original baseline version, to the effect that it is once again in a close second group position. The dependency-based method fares much better in the case of German data, and ends up in the second best-performing group together with the smoothed BLEU. TER, on the other hand, shows a considerably lower correlation than any other metric, including the original BLEU score. It also falls far behind its

own result on the French data, which suggests that its performance is influenced to a large extent by the character of the input language.

7.3 Spanish

The Spanish section is twice as large as the remaining target language sets and includes 15,919 translation-reference segments with associated human scores. As before, we evaluated it with the same set of metrics, calculated the correlations with human judgments and the statistical significance of differences in correlations. Table 7.4 presents the results.

Spanish	
BLEU_s, METEOR_st	0.595∗
METEOR_e	0.593∗
NIST	0.557
d_pm	0.534†
d	0.530†
GTM	0.527†
TER	-0.504
BLEU	0.453

Table 7.4: Correlations with average human judgment for Spanish. Legend: **METEOR_st** = METEOR with stemming; **METEOR_e** = METEOR in "exact" mode; **BLEU_s** = BLEU with add-one smoothing; **d** = dependency-based method; **d_pm** = dependency-based method with partial matching; ∗ and † indicate scores that are not significantly different from each other.

The correlations calculated over the Spanish data show similar effects to the French subset, although this time BLEU with add-one smoothing catches up with METEOR as the best-performing metric. The dependency-based method does slightly better than in the French case, and together with GTM finds itself in third position. However, unlike during the evaluation of the German data, it is overtaken by NIST. TER and the original BLEU again close the ranking, although TER's correlation is considerably stronger than

for German. An interesting feature of the Spanish data set is that the span between the best- and worst-performing metrics that were used to evaluate the data is particularly low here; it measures 0.142 points, in contrast to the French (0.221), German (0.264), and Japanese data (0.393).

The Spanish data set made it possible to carry out a comparison of the dependencies generated by NLPWin with those produced by a broad-coverage Spanish LFG parser (Chrupała and van Genabith (2006a,b)), and their respective performance in MT evaluation. We processed the Spanish translation and reference segments with the LFG parser, producing dependencies of the kind described in Section 6.3. Then we used the dependencies to evaluate the Spanish data set, conducting the dependency-based evaluation in the baseline version and with partial matching, as in all experiments presented here. Table 7.5 repeats the NLPWin-produced dependency correlations from Table 7.4 and compares them with the correlations obtained on the same data by the LFG dependencies.

	Spanish
lfg_d	0.537∗
d_pm	0.534∗
lfg_d_pm	0.531∗
d	0.530∗

Table 7.5: Correlations with average human judgment for Spanish for two dependency sources. Legend: **d** = dependency-based method using dependencies from NLPWin; **lfg_d** = dependency-based method using LFG dependencies; **_pm** = partial matching; ∗ indicate scores that are not significantly different from each other.

It seems clear that the origin of dependencies has no influence in this case on the method's quality. None of the differences are significant, and the correlation values are very close to one another. We can therefore conclude that both NLPWin and the Spanish LFG parser produce dependencies that are equally suited to the task of MT evaluation, in contrast to the CCG parser employed by Mehay and Brew (2007).

7.4 Japanese

The Japanese data set was the most problematic of all the target languages provided by Microsoft. Since the language is not Indo-European, the behaviour of the evaluation metrics, designed mostly for Western languages, was not predictable. It also meant that some metric options were not available. For instance, in the case of METEOR only the exact match mode could be used, since METEOR only provides stemming for English, Spanish, German, and French. However, one could argue that the morphological segmentation of the Japanese text carried out in the pre-processing stage with the help of the JUMAN analyzer is, to a certain extent, similar to the stemming performed by METEOR.

Japanese	
BLEU_s	0.559
METEOR_e	0.535
d_pm	0.495∗
d	0.489∗
NIST	0.466
BLEU	0.429
TER	-0.166

Table 7.6: Correlations with average human judgment for Japanese. Legend: **METEOR_e** = METEOR in "exact" mode; **BLEU_s** = BLEU with add-one smoothing; **d** = dependency-based method; **d_pm** = dependency-based method with partial matching; ∗ indicates scores that are not significantly different from each other.

This time, smoothed BLEU and METEOR show the highest correlations with the human scores, with BLEU achieving a significantly better result. Both versions of the dependency-based method obtain a similar result and are in turn significantly better than NIST, the original BLEU, and TER. Interestingly, the overall correlations for segmented Japanese input are for the most part comparable to those for the Western languages explored earlier, which suggests the universal applicability of the metrics. The clear

exception here is TER with its very low correlation of -0.166. Another exception is GTM, which did not work for this data set at all (although this is probably just a matter of program implementation).

7.5 The four languages: a summary

The experiments on the four languages show results that are largely consistent with the experiments carried out for the English data sets: a version of the dependency-based method without n-best parses or WordNet performs reasonably well in comparison to other metrics, although it lags behind METEOR and smoothed BLEU in most cases. For French and Spanish, it is outperformed by NIST as well. The French test set seems particularly challenging for our method; the method achieves correlations on a par with TER and only slightly higher than the original version of BLEU, both of which are consistently the weakest contenders in all comparisons. One possible reason for such localized difficulty might be that NLPWin produces weaker quality output for French than for other languages; however, this remains speculation as no evaluation data for NLPWin is available. The only supporting evidence is that the ranking for the remaining metrics in French is very similar to the other rankings for German, Spanish, and Japanese, so whatever causes the poor performance of the dependency-based method is likely to be caused by something inherent to the method, e.g. the parse.

German and Japanese, on the other hand, turn out to be extremely problematic for TER. In Japanese, the added difficulty might be that the input had been segmented into morphemes, and TER was designed to work on fully inflected words. It is not clear why TER has trouble with the German translations; at first sight there seems to be nothing in particular about the data that distinguishes it from the other languages. The average number of words in a segment is similar: 16 for German, 16.7 for French, and 18.6 for Spanish. Japanese stands out here with the count of 21, but we count morphemes

instead of words. The average number of edits performed by TER per sentence is, however, slightly higher for German and Japanese: 10.8 and 13.4, respectively, whereas TER makes on average 9 edits in a French segment, and 7.8 in a Spanish segment. As a result, German and Japanese experience the largest percentage of TER edits per sentence: 62.5% in German and 63.8% in Japanese, in contrast to French 53.9% and Spanish 41.9%.

The percentage of edits might not tell us much about the root of the problem. Perhaps the German data is of worse quality than the other language samples, and the higher number of edits is necessary. However, the average human score for German is 2.26, whereas for Japanese it is 2.51, for French 2.36, and for Spanish 2.74. Even though the average German score is the lowest, it is only slightly lower than the average score for French, a language in which TER shows its usual quality, suggesting that the quality of the German translation cannot be the reason for TER's poor performance in this sample.

The highest correlations with human scores are obtained in all cases by METEOR or BLEU with add-one smoothing. Where both exact and combined (i.e. by "combined" we mean exact and stemmed) versions of METEOR were applied, the latter performed better, although only in the case of French the difference is statistically significant. As mentioned before, BLEU with add-one smoothing owes its good results to the fact that the addition of one to every count across-the-board (increasing the number of counted matches) counteracts the systematic underestimation of translation quality on segment level which is the main weakness of the original BLEU.

NIST and GTM invariably find themselves in the middle of the range, with NIST significantly outperforming GTM twice, in the German and Spanish data sets (note again that we were not able to produce a GTM score for the Japanese set).

The results obtained by the dependency-based method in these four comparisons are clearly not as good as those achieved in the earlier chapters where English was

the target language. However, many of the improvements to the method discussed in Chapter 5 were not available for these experiments, due to different target languages and the character of dependencies used. In effect, we were only able to employ the basic version of the method and enhance it with partial matching. Even if it does not obtain the highest correlations, the method performs in a rather consistent manner throughout the languages, which cannot be said for TER, for instance. It also shows similar results whether it relies on the dependencies produced by an LFG parser or those generated by NLPWin.

Chapter 8

Conclusions

The increasing amount of research in the area of Machine Translation and the evolving sophistication and quality of the translation models mean that there is a growing need for reliable automatic evaluation methods, able to mirror faithfully the quality of the unfortunately slow and expensive process of human assessment. To date, most popular automatic evaluation metrics have relied on a surface comparison of the translation and reference on a string level, essentially limiting themselves to calculating how many words are shared between the two strings. Even given reordering, stemming, and synonyms for individual words, current methods still lag far behind human ability to assess the quality of translation, as is evident in the correlation studies (e.g. Callison-Burch et al. (2006b, 2007)). It is becoming clear that string-based comparison, although extremely useful and simple to understand and apply, might have exhausted its potential, especially in the case of segment-level evaluation. As the quality of MT technology improves, string matches are no longer sophisticated enough to distinguish more subtle variation in translation quality, and there is a need for a deeper linguistic analysis of translated text. On the other hand, one needs to keep in mind the potential unreliability of human evaluation in the comparisons of human and automatic metrics, especially when it is performed on a small scale with only one or two judgments per segment.

Our method explores the direction of deeper linguistic analysis in MT evaluation by comparing sentences on the level of their local grammatical relations, as exemplified by their f-structure dependency triples produced by an LFG parser or by the NLPWin analyzer. The dependency-based method can be further augmented by allowing partial matching for predicate dependencies and adding WordNet synonyms, which help account for potential lexical divergences between translation and reference segments. While the dependency-based method does not provide a full semantic analysis of the translation (which would further approximate the ideal "human-level" evaluation), it produces a shallow breakdown of the segments in terms of predicates, subjects, objects, and grammatical information such as number, person, etc. In our experiments on English data we showed that the dependency-based method correlates higher than any other metric with human evaluation of translation fluency and with the average human score on a segment level.

On the system level, the method is outperformed by BLEU and METEOR when it comes to correlations with human judgments of fluency and adequacy, respectively, but it obtains the highest correlation with the average human score. When the system-level correlations are calculated using approximate randomization and clustering to test for statistical significance, as in Stroppa and Owczarzak (2007), METEOR takes the lead in all three categories, with the dependency-based method in the second place.

By examining several versions of our method we show that it has the potential for further improvement; we also demonstrate that some of these modifications are better suited to MT evaluation than others. For example, unordered sets of local dependencies seem to work better than long chains inspired by tree paths such as those by Liu and Gildea (2005). At the same time, our method proves to be one of the least biased towards the statistical models of translation, and as such provides a more reliable and universal evaluation, especially given that SMT is by far the most dominant paradigm in MT system development today.

Naturally, since our method requires an external resource such as a reliable dependency parser for the target language, its usability is currently confined to those languages for which such resources exist. Fortunately, more and more languages, including non-Indo-European languages, have started to receive the attention of NLP researchers in recent years, and new tools for these languages are being developed on a regular basis. The suitability of our method for evaluation in languages other than English is shown in the experiments on Microsoft's dependency structures for Spanish, French, German, and Japanese, produced by the NLPWin analyzer of Corston-Oliver and Dolan (1999) and Heidorn (2000). While its performance in these languages is not as good as it was for English, the method appears to be rather consistent in its cross-linguistic quality of evaluation, contrary to, for example, TER.

We also suggest that output of some parsers might be more appropriate for the task of MT evaluation than others. In the experiment on the Spanish data, the dependencies produced by the LFG parser of Chrupała and van Genabith (2006a,b) and those generated by Microsoft's NLPWin analyzer provided similar quality evaluation when used in the dependency-based method. In English, however, LFG-generated dependencies seem to fare better in MT evaluation than those produced by the CCG parser in Mehay and Brew (2007).

One of the important questions we must ask if we want to see further improvements to MT evaluation methods is: what exactly are we trying to model? What is our 'gold standard'? Not only is human judgment inherently variable, and most of today's evaluation tasks do not employ adequate countermeasures for this variability, but also the standard human evaluation scales might inadvertently contribute to the problem with the design and reliable comparison of automatic metrics. Fluency and adequacy are two very different aspects of translation quality, each with its own set of conditions along which the input is evaluated. Therefore, it seems unfair to expect a single automatic metric to correlate highly with human judgments of both fluency and adequacy at the

same time. Yet, this is how most research on automatic metrics is carried out today. In Chapter 7 we saw that results improve when automatic scores are correlated with a single human score denoting the overall translation quality. This suggests that we either reconsider the design of human judgment studies, or, if we choose to keep the existing values of fluency and adequacy, that we target these correlations separately, if we want our automated metrics to reflect human scores better.

In order to better model human judgments of adequacy, an important step would be to push the automatic evaluation of MT quality towards including an even deeper-level linguistic analysis, where a semantic comparison between a translation segment and a reference could be performed, perhaps by using semantic role labelling coupled with synonym detection. On the other hand, to reflect human fluency judgments, we could employ syntactic language models on the lines of Rajman and Hartley (2001) and the early evaluation studies, where intelligibility (fluency) was assessed as a quality unrelated to the content and form of a reference text.

These developments would bring the automatic evaluation metrics much closer to human assessment in terms of the evaluation process, and, hopefully, in terms of the evaluation quality as well. As better quality will be rearranged with certitude on better quality of automatic translation technician of estimate, perhaps, this opinion (sentence) will be intelligible (apprehended) in future without check translations.[1]

[1]Reference: *And since a better evaluation quality will surely lead to a better quality of Machine Translation itself, this sentence might one day be understandable without a reference.*

Bibliography

Akiba, Y., Imamura, K., and Sumita, E. (2001). Using multiple edit distances to automatically rank Machine Translation output. In *Proceedings of MT Summit 2001*, pages 15–20, Santiago de Compostela, Spain.

Albrecht, J. S. and Hwa, R. (2007). Regression for sentence-level MT evaluation with pseudo references. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 296–303, Prague, Czech Republic.

Banerjee, S. and Lavie, A. (2005). METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL 2005 Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization*, pages 65–73, Ann Arbor, MI, USA.

Bannard, C. and Callison-Burch, C. (2005). Paraphrasing with bilingual parallel corpora. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 597–604, Ann Arbor, MI, USA.

Bikel, D. M. (2002). Design of a multilingual, parallel-processing statistical parsing engine. In *Proceedings of Human Language Technology Conference 2002*, pages 24–27, San Diego, CA, USA.

Bresnan, J. (2001). *Lexical-Functional Syntax*. Blackwell, Oxford.

Cahill, A., Burke, M., O'Donovan, R., Riezler, S., van Genabith, J., and Way, A. (2008). Wide-coverage deep statistical parsing using automatic dependency structure annotation. *To appear in: Computational Linguistics*.

Cahill, A., Burke, M., O'Donovan, R., van Genabith, J., and Way, A. (2004). Long-distance dependency resolution in automatically acquired wide-coverage PCFG-based

LFG approximations. In *Proceedings of the 42th Annual Meeting of the Association of Computational Linguistics*, pages 320–327, Barcelona, Spain.

Callison-Burch, C., Fordyce, C., Koehn, P., Monz, C., and Schroeder, J. (2007). (Meta-)evaluation of Machine Translation. In *Proceedings of the ACL 2007 Workshop on Statistical Machine Translation*, pages 136–158, Prague, Czech Republic.

Callison-Burch, C., Koehn, P., and Osborne, M. (2006a). Improved Statistical Machine Translation using paraphrases. In *Proceedings of Human Language Technology - North American Chapter of the Association for Computational Linguistics*, pages 17–24, New York, NY, USA.

Callison-Burch, C., Osborne, M., and Koehn, P. (2006b). Re-evaluating the role of BLEU in Machine Translation research. In *Proceedings of the 11th Conference of the European Chapter of the Association of Computational Linguistics*, pages 249–256, Oslo, Norway.

Charniak, E. (2000). A maximum entropy inspired parser. In *Proceedings of the 1st Annual Meeting of the North American Chapter of the Association for Computational Linguistics*, page 132139, Seattle, WA, USA.

Chrupała, G. (2006). Simple data-driven context-sensitive lemmatization. In *Proceedings of 22th Annual Meeting of the Sociedad Española para el Procesamiento del Lenguaje Natural*, pages 121–127, Zaragoza, Spain.

Chrupała, G. and van Genabith, J. (2006a). Improving treebank-based automatic LFG induction for Spanish. In *Proceedings of the LFG06 Conference*, pages 91–106, Konstanz, Germany.

Chrupała, G. and van Genabith, J. (2006b). Using machine-learning to assign function labels to parser output for Spanish. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 136–143, Sydney, Australia.

Clark, S. and Curran, J. R. (2004). Parsing the WSJ using CCG and log-linear models. In *Proceedings of the 42th Annual Meeting of the Association of Computational Linguistics*, pages 103–111, Barcelona, Spain.

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20:3746.

Collins, M. (1999). *Head-Driven Statistical Models for Natural Language Parsing*. PhD thesis, University of Pennsylvania, Philadelphia, PA, USA.

Corston-Oliver, S. H. and Dolan, W. B. (1999). Less is more: Eliminating index terms from subordinate clauses. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 349–356, College Park, MD, USA.

Diab, M. and Resnik, P. (2002). An unsupervised method for word sense tagging using parallel corpora. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 255–262, Philadelphia, PA, USA.

Doddington, G. (2002). Automatic evaluation of MT quality using n-gram co-occurrence statistics. In *Proceedings of Human Language Technology Conference 2002*, pages 138–145, San Diego, CA, USA.

Dolan, W. B., Pinkham, J., and Richardson, S. D. (2002). MSR-MT: The Microsoft Research Machine Translation system. In *Proceedings of the 5th Conference of the Association for Machine Translation in the Americas on Machine Translation*, pages 237–239, Tiburon, CA, USA.

Fisher, R. A. (1990). *Statistical Methods, Experimental Design, and Scientific Inference: A Re-issue of Statistical Methods for Research Workers, The Design of Experiments, and Statistical Methods and Scientific Inference*. Oxford University Press, USA.

Fleiss, J. L. (1971). Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76:5:378–382.

Giménez, J. and Màrquez, L. (2007). Linguistic features for automatic evaluation of heterogeneous MT systems. In *Proceedings of the ACL 2007 Workshop on Statistical Machine Translation*, pages 256–264, Prague, Czech Republic.

Green, T. (1979). The necessity of syntax markers. Two experiments with artificial languages. *Journal of Verbal Learning and Behavior*, 18:481–496.

Groves, D. and Way, A. (2005). Hybrid Example-Based SMT: The best of both worlds? In *Proceedings of the ACL 2005 Workshop on Building and Using Parallel Texts: Data-Driven Machine Translation and Beyond*, pages 183–190, Ann Arbor, MI, USA.

Halkidi, M., Batistakis, Y., and Vazirgiannis, M. (2001). On clustering validation techniques. *Journal of Intelligent Information Systems*, 17(2-3):107–145.

Heidorn, G. E. (2000). *Handbook of Natural Language Processing*, chapter Intelligent Writing Assistance. CRC Press, USA.

Hockenmaier, J. (2003). Parsing with generative models of predicate-argument structure. In *Proceedings of the 41th Annual Meeting of the Association of Computational Linguistics*, pages 359–366, Sapporo, Japan.

Kaplan, R. M. and Bresnan, J. (1982). *The Mental Representation of Grammatical Relations*, chapter Lexical-Functional Grammar: A Formal System for Grammatical Representation. MIT Press, Cambridge, MA, USA.

Kauchak, D. and Barzilay, R. (2006). Paraphrasing for automatic evaluation. In *Proceedings of Human Language Technology North American Chapter of the Association of Computational Linguistics Conference 2006*, pages 455–462, New York, NY, USA.

Koehn, P. (2004). Pharaoh: A beam search decoder for phrase-based Statistical Machine Translation models. In *Proceedings of the Workshop on Machine Translation: From real users to research at the Association for Machine Translation in the Americas Conference 2004*, pages 115–124, Washington, DC, USA.

Koehn, P. (2005). Europarl: A parallel corpus for Statistical Machine Translation. In *Proceedings of MT Summit 2005*, pages 79–86, Phuket, Thailand.

Kulesza, A. and Shieber, S. M. (2004). A learning approach to improving sentence-level MT evaluation. In *Proceedings of the 10th Conference on Theoretical and Methodological Issues in Machine Translation*, pages 75–84, Baltimore, MD, USA.

Landis, R. J. and Koch, G. K. (1977). The measurement of observer agreement for categorical data. *Biometrics*, 33:159–174.

Lavie, A. and Agarwal, A. (2007). Meteor: An automatic metric for MT evaluation with high levels of correlation with human judgments. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 228–231, Prague, Czech Republic.

LDC (2005). Linguistic data annotation specification: Assessment of fluency and adequacy in translations. Revision 1.5.

Leusch, G., Ueffing, N., and Ney, H. (2003). A novel string-to-string distance measure with applications to Machine Translation evaluation. In *Proceedings of MT Summit 2003*, pages 240–247, New Orleans, LO, USA.

Leusch, G., Ueffing, N., and Ney, H. (2006). CDER: Efficient MT evaluation using block movements. In *Proceedings of the 11th Conference of the European Chapter of the Association of Computational Linguistics*, pages 241–248, Trento, Italy.

Levenshtein, V. (1966). Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10:707–710.

Lin, C.-Y. and Och, F. J. (2004). Automatic evaluation of Machine Translation quality using longest common subsequence and skip-bigram statistics. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, pages 605–612.

Liu, D. and Gildea, D. (2005). Syntactic features for evaluation of Machine Translation. In *Proceedings of the ACL 2005 Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 25–32, Ann Arbor, MI, USA.

Magerman, D. M. (1995). Statistical decision-tree models for parsing. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, pages 276–283, Cambridge, MA, USA.

Mehay, D. N. and Brew, C. (2007). BLEUÂTRE: Flattening syntactic dependencies for MT evaluation. In *Proceedings of the 11th International Conference on Theoretical and Methodological Issues in Machine Translation*, pages 122–131, Skoevde, Sweden.

Melamed, I. D., Green, R., and Turian, J. P. (2003). Precision and recall of Machine Translation. In *Proceedings of Human Language Technology North American Chapter of the Association of Computational Linguistics Conference 2003*, pages 61–63, Edmonton, Canada.

Mellebeek, B. (2007). *TransBooster: Black Box Optimisation of Machine Translation Systems*. PhD thesis, Dublin City University, Dublin, Ireland.

Mellebeek, B., Khasin, A., van Genabith, J., and Way, A. (2005a). Transbooster: Boosting the performance of wide-coverage Machine Translation systems. In *Proceedings of*

the 10th Conference of the European Association for Machine Translation, pages 189–197, Budapest, Hungary.

Mellebeek, B., Owczarzak, K., Groves, D., van Genabith, J., and Way, A. (2006a). A syntactic skeleton for Statistical Machine Translation. In *Proceedings of the 11th Conference of the European Association for Machine Translation*, pages 195–202, Oslo, Norway.

Mellebeek, B., Owczarzak, K., Khasin, A., van Genabith, J., and Way, A. (2005b). Improving online Machine Translation systems. In *Proceedings of MT Summit 2005*, pages 290–297, Phuket, Thailand.

Mellebeek, B., Owczarzak, K., van Genabith, J., and Way, A. (2006b). Multi-engine Machine Translation by recursive sentence decomposition. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas*, pages 110–118, Cambridge, MA, USA.

Niessen, S., Och, F. J., Leusch, G., and Ney, H. (2000). An evaluation tool for Machine Translation: Fast evaluation for MT research. In *Proceedings of the 2nd International Conference on Language Resources and Evaluation*, pages 39–45, Athens, Greece.

Noreen, E. W. (1989). *Computer-intensive methods for testing hypotheses: An introduction.* Wiley-Interscience, New York, NY, USA.

Och, F. J. and Ney, H. (2003). A systematic comparison of various statistical alignment modes. *Computational Linguistics*, 29:19–51.

Owczarzak, K., Graham, Y., and van Genabith, J. (2007a). Using f-structures in Machine Translation evaluation. In *Proceedings of the LFG07 Conference*, pages 383–396, Stanford, CA, USA.

Owczarzak, K., Groves, D., van Genabith, J., and Way, A. (2006a). Contextual bitext-derived paraphrases in automatic MT evaluation. In *Proceedings of the HLT-NAACL 2006 Workshop on Statistical Machine Translation*, pages 86–93, New York, NY, USA.

Owczarzak, K., Mellebeek, B., Groves, D., van Genabith, J., and Way, A. (2006b). Wrapper syntax for Example-Based Machine Translation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas*, pages 148–155, Cambridge, MA, USA.

Owczarzak, K., van Genabith, J., and Way, A. (2007b). Dependency-based automatic evaluation for Machine Translation. In *Proceedings of the HLT-NAACL 2007 Workshop on Syntax and Structure in Statistical Machine Translation*, pages 86–93, Rochester, NY, USA.

Owczarzak, K., van Genabith, J., and Way, A. (2007c). Labelled dependencies in Machine Translation evaluation. In *Proceedings of the ACL 2007 Workshop on Statistical Machine Translation*, pages 104–111, Prague, Czech Republic.

Owczarzak, K., van Genabith, J., and Way, A. (2008). Evaluating Machine Translation with LFG dependencies. *To appear in: Machine Translation.*

Palmer, M., Gildea, D., and Kingsbury, P. (2005). The Proposition Bank: An annotated corpus of semantic roles. *Computational Linguistincs*, 31(1):71–106.

Pang, B., Knight, K., and Marcu, D. (2003). Syntax-based alignment of multiple translations: Extracting paraphrases and generating new sentences. In *Proceedings of Human Language Technology - North American Chapter of the Association for Computational Linguistics 2003*, page 181188, Edmonton, Canada.

Papineni, K., Roukos, S., Ward, T., and Zhu, W. (2002). BLEU: A method for automatic evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association of Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania.

Paul, M. (2006). Overview of the IWSLT 2006 evaluation campaign. In *Proceedings of the International Workshop on Spoken Language Translation*, pages 1–15, Kyoto, Japan.

Pierce, J. R., Carroll, J. B., Hamp, E. P., Hays, D. G., Hockett, C. F., Oettinger, A. G., and Perlis, A. (1966). Language and machines computers in translation and linguistics. Technical report, Automatic Language Processing Committee, National Academy of Sciences, National Research Council, Washington, DC, USA.

Rajman, M. and Hartley, T. (2001). Automatically predicting MT systems rankings compatible with fluency, adequacy or informativeness scores. In *Proceedings of MT Summit 2001*, pages 29–34, Santiago de Compostela, Spain.

Richardson, S. D. (2004). Machine translation of online product support articles using a data-driven MT system. In *Proceedings of the Workshop on Machine Translation: From real users to research at the Association for Machine Translation in the Americas Conference 2004*, pages 246–251, Washington, DC, USA.

Riezler, S. and Maxwell, J. (2005). On some pitfalls in automatic evaluation and significance testing for MT. In *Proceedings of the ACL 2005 Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization*, pages 57–64, Ann Arbor, MI, USA.

Russo-Lassner, G., Lin, J., and Resnik, P. (2005). A paraphrase-based approach to Machine Translation evaluation. Technical Report Technical Report LAMP-TR-125/CS-TR-4754/UMIACS-TR-2005-57, University of Maryland, College Park, MD, USA.

Snover, M., Dorr, B., Schwartz, R., Makhoul, J., and Micciula, L. (2006). A study of translation error rate with targeted human annotation. In *Proceedings of the Association for Machine Translation in the Americas Conference 2006*, pages 223–231, Boston, MA, USA.

Stroppa, N. and Owczarzak, K. (2007). A cluster-based representation for multi-system MT evaluation. In *Proceedings of the 11th International Conference on Theoretical and Methodological Issues in Machine Translation*, pages 221–230, Skoevde, Sweden.

Turian, J. P., Shen, L., and Melamed, I. D. (2003). Evaluation of Machine Translation and its evaluation. In *Proceedings of MT Summit 2003*, pages 386–393, New Orleans, LA, USA.

van Slype, G. (1979). Critical methods for evaluating the quality of Machine Translation. Prepared for the European Commission Directorate General Scientific and Technical Information and Information Management. Technical Report BR-19142, Bureau Marcel van Dijk.

White, J. S., O'Connell, T., and O'Mara, F. (1994). The ARPA MT evaluation methodologies: Evolution, lessons and further approaches. In *Proceedings of the 1st Conference of the Association for Machine Translation in the Americas*, pages 193–205, Columbia, MD, USA.

Ye, Y., Zhou, M., and Lin, C.-Y. (2007). Sentence level Machine Translation evaluation as a ranking problem: One step aside from BLEU. In *Proceedings of the ACL 2007 Workshop on Statistical Machine Translation*, pages 240–247, Prague, Czech Republic.