Diplomarbeit im Fach Informatik
RHEINISCH-WESTFÄLISCHE TECHNISCHE HOCHSCHULE AACHEN
Lehrstuhl für Informatik VI
Prof. Dr.-Ing. H. Ney

Evaluation Measures in Machine Translation

vorgelegt von:
Cand. Inf. Gregor Leusch
Matrikelnr. 216.502

Gutachter:
Prof. Dr.-Ing. H. Ney
Prof. Dr. T. Seidl

Betreuerin:
Dipl.-Math. N. Ueffing

Hiermit versichere ich, dass ich die vorliegende Diplomarbeit selbständig verfasst und keine anderen als die angegebenen Hilfsmittel verwendet habe. Alle Textauszüge und Grafiken, die sinngemäß oder wörtlich aus veröffentlichten Schriften entnommen wurden, sind durch Referenzen gekennzeichnet.

Aachen, im Juli 2005

Gregor Leusch

Acknowledgments

The present work originates from my research as student worker at the Chair of Computer Science VI of the RWTH Aachen University, where I have been a student member of the machine translation group for a few years. I would like to thank Prof. Dr.-Ing. Hermann Ney for the interesting research possibilities at his department, including the chance to submit two research papers to international conferences.

I would like to thank Prof. Dr. Thomas Seidl as well, who kindly accepted to attend the thesis.

Moreover, I would like to thank the members of the Machine Translation group, namely Oliver Bender, Saša Hasan, Evgeny Matusov, Sonja Nießen, Maja Popović, David Vilar, Jia Xu, Richard Zens, and most of all, Nicola Ueffing, for their advice, hints, discussions, suggestions, and much more that helped creating this work.

Furthermore, I would like to thank my fellow graduate candidates and student workers — especially Andre Altmann, Ilja Bezrukov, Stefan Hahn, Andre Hegerath, Arne Mauser, Thomas Scharrenbach, Thomas Schoenemann, and Daniel Stein — for many helpful discussions, comments, and hints.

Not to forget my friends from the Filmstudio: Thank you – just for being friends.

And of course special thanks go to my parents and my siblings for always being there when I needed them.

Abstract

Continual evaluation of the translation quality is an essential part of machine translation (MT) research. Consequently, there is a strong need for quick and reliable evaluation methods in this field. This diploma thesis addresses the automatic evaluation of machine translation. The evaluation methods investigated in this thesis are based on the automatic, numerical comparison of reference translations with translations generated by the MT systems. After an introduction to the subject, several string based similarity and distance measures will be presented.

In addition to the well-established measures WER, PER, NIST, and BLEU, three newly developed distance measures will be introduced. The first new distance measure is based on m-gram or skip bigram count vectors. Basically, it is an combination of the PER distance principle with BLEU m-gram count vectors. Additionally, two novel distance measures are based on edit operations including block reordering. Previous block reordering measures are based on NP-hard problems and can thus only be calculated approximately in the general case. The new block reordering measures presented here are respectively based on bracketing transduction grammar restrictions to the set of possible permutations and on the drop of certain coverage constrains. Therefore, these new measures can be calculated in polynomial time.

Further topics of this thesis are different preprocessing methods for automatic MT evaluation, the handling of sentence boundaries, methods for the determination of reference lengths, and methods for the evaluator normalization. All these topics have in common that they are relevant for a multitude of automatic evaluation measures.

The pivotal quality benchmark for automatic MT evaluation measures is the correlation between them and human evaluation. Consequently, all measures and methods in this work will be compared experimentally by calculating this correlation. These experiments are conducted on data of seven international MT evaluation campaigns. On all corpora, the novel evaluation measures and methods presented in the thesis increase the correlation between automatic and human evaluation by 24 to 45 percent relative in comparison with baseline BLEU.

Zusammenfassung

Die stetige Bewertung der Übersetzungsqualität ist essentieller Part des Forschungsgebietes maschineller Übersetzungen (MT). Dementsprechend besteht in diesem Gebiet ein deutlicher Bedarf an schnellen und zuverlässigen Bewertungsmethoden. Diese Diplomarbeit behandelt die automatische Bewertung maschineller Übersetzungssysteme. Die hierbei untersuchten Methoden beruhen auf dem Vergleich von Referenzübersetzungen mit von den Systemen generierten Testübersetzungen. Nach einer Einführung in die Thematik werden verschiedene zeichenkettenbasierte Ähnlichkeits- und Abstandsmaße vorgestellt.

Zusätzlich zu den etablierten Maßen WER, PER, NIST und BLEU werden drei neue Abstandsmaße eingeführt. Das erste dieser Maße beruht auf m-gram- oder Skip Bigram-Zählvektoren. Im wesentlichen ist es eine Kombination der Abstands-Prinzipien der PER mit den m-gram-Zählvektoren von BLEU. Die weiteren Maße basieren auf Editieroperationen mit Blockvertauschungen. Bisherige Blockvertauschungs-Distanzmaße beruhen auf NP-harten Problemen und lassen sich daher im allgemeinen Fall nur approximativ lösen. Die beiden neuen Maße beinhalten im einen Fall auf Beschränkungen durch Bracketing Transduction Grammars an die Menge der möglichen Vertauschungen, im anderen Fall Lockerungen der Abdeckungs-Bedingung. Beide Maße können dadurch in Polynomzeit berechnet werden.

Weitere Themen dieser Diplomarbeit sind die verschiedenen Vorverarbeitungsmethoden innerhalb der automatischen Bewertung maschineller Übersetzungen, die Behandlung von Satzgrenzen, Methoden zur Referenzlängenbestimmung sowie Methoden zur Bewerternormalisierung. Allen diesen Punkten ist gemeinsam, dass sie für eine Vielzahl verschiedener Bewertungsmaße relevant sein können.

Der entscheidende Qualitätsmaßstab für automatische Bewertungsmaße in der maschinellen Übersetzung ist die Korrelation dieser Maße mit menschlicher Bewertung. Dementsprechend werden die vorgestellten Bewertungsmaße und Methoden experimentell auf den Daten sieben internationaler Evaluierungskampagnen hinsichtlich der Korrelation mit menschlicher Bewertung untersucht. Es ergibt sich, dass mit den vorgestellten neuen Maßen und Methoden diese Korrelation auf allen Korpora um 24% bis 45% relativ gegenüber dem ursprünglichen BLEU verbessert werden kann.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

This chapter gives a short introduction to the Machine Translation research process, explains why evaluation of Machine Translation is essential here, and lists the topics this thesis will cover.

1.1 Machine Translation

With the globalization of business and the decline of national borders, the importance of international communication raises continuously. A major hindrance for communication is the vast number of languages. Consequently, there is an enormous need for translation between these different languages. For day-to-day communication, **Machine Translation** (MT), the automatic translation of speech or written text among natural languages, has begun to complement human translators, a process that is expected to continue for the next years.

The task of an MT system is clear: The system receives a sentence or text in one natural language, the **source language**, as user input. It translates the source sentence into another natural language, the **target language**. This target sentence is then returned to the user. In spite of this simple description, MT as a research subject has seen an abundance of approaches and ideas — for example, linguistic, computer-linguistic, rule-based, example-based, statistical, and many more. Among these, **Statistical Machine Translation** (SMT) has proved its practical leadership in many competitions and evaluation campaigns. Consequently, most of the experiments for this work have been conducted on SMT output, although the methods presented here will be valid for non–statistical MT systems as well.

1.2 Evaluation in Machine Translation

As any other task in natural language processing (NLP), MT research depends on continual evaluation. The large amount of MT approaches demands for system independent comparison of different approaches with regard to their quality. Different implementations of the same approach need to be compared as well. System parameters, especially

those from SMT systems, must be optimized. For all this, a method to assess the quality of an MT system is required. Over the last years, a manifold of evaluation measures has been proposed and studied for this purpose. This underlines the importance, but also the complexity of finding a suitable evaluation measure for MT.

Generally, evaluation and comparison of MT systems takes place by sending a fixed **test set** of source language sentences to the systems. These sentences should come from the same domain the MT systems were trained on. Then, the MT systems translate these source sentences into the target language. The generated sentences, called **candidate translations**, are then assessed. Evaluation scores can be calculated on the level of whole test sets, as well as on the level of single test sentences. The former is the method of choice to compare different MT approaches, or to automatically adjust parameters. The latter is useful when the actual effects of a certain change in MT system parameters have to be analyzed.

1.2.1 Human evaluation

The most obvious way to assess translation quality is to have human evaluators mark candidate translations. Marks can be given for different aspects of translation quality, or as an overall score. In recent international evaluation campaigns (e.g., [LDC 05, Akiba & Federico$^+$ 04]), the only assessed translation quality aspects have been syntactical quality and semantical quality.

The syntactical quality, usually called **fluency**, describes the readability and understandability of a sentence to the human reader, independently of the semantics. This is a monolingual feature, and thus a monolingual evaluator can assess it.

The semantical quality, usually called **adequacy**, describes the correctness of the conveyed information in the candidate sentence. As this correctness depends highly on the information contained in the corresponding source sentence, either bilingual evaluators must undertake this evaluation, or a monolingual evaluator must compare the candidate translations with appropriate reference translations.

The main advantage of human MT evaluation over automatic MT evaluation is that human users are the ones who have to read and use MT system output in practice*. As these human users know best what they can understand and what the semantical value of a natural language sentence is, we can expect them to judge the practical usability and appropriateness of MT more precisely than any computer could do.

On the other hand, human MT evaluation is a rather time-consuming task: Both [Nießen & Och$^+$ 00] and TIDES [LDC 05] give an estimate of about 30 seconds per evaluated sentence, multiplied by the number of evaluators the sentence is assessed by. Moreover, a test set usually consists of several hundred or thousand sentences, and an evaluation campaign can comprise some five to fifteen MT systems, each with its own set of candidate translations. Therefore human evaluation in a campaign will cost a large amount of person hours for human experts. In consequence, human evaluation is also very costly in financial aspects.

*Cross-language information retrieval and similar tasks being an exception here.

Furthermore, provisions must be taken to avoid or cancel out a possible bias of the human evaluators. This can be done by the means of a database, where a single evaluator or a group of evaluators will always see the scores of previously evaluated sentences. This gives them the possibility to reevaluate old scores, such that the relative ranking of the sentences is unbiased afterwards. [Nießen & Och$^+$ 00] give a description of this method. Although a temporary bias consequently does not perturb a local ranking, the database method cannot avert the possible case where an evaluator prefers the "style" of certain MT systems towards the style of the other systems. A more popular approach to deal with bias is to distribute the candidate sentences randomly to several human evaluators such that at least two judges evaluate each sentence independently from one another. All scores of a sentence are then averaged. This procedure cancels out all biases on system level. However, it does not necessarily take effect on sentence level, especially if there are only few assessments per candidate sentence.

Still, human evaluation basically is subjective evaluation. Therefore, even with the specified procedures, subjective evaluation results are not necessarily reproducible by different groups of human evaluators. Reproducibility is not even guaranteed for the same group of evaluators at a later time.

1.2.2 Automatic evaluation

The high evaluation costs of human evaluation are a serious problem, especially for the evaluation for SMT parameter optimization, where dozens or hundreds of evaluation runs are conducted on the same test set. Although only a limited set of candidate translations will change after each small-scale parameter change, there is still a large amount of assessments necessary for an average set of parameter training iterations. The low reproducibility of human evaluation on the other hand is especially problematic when evaluating new approaches or systems.

To overcome the problems of high evaluation costs on the one hand and of the low reproducibility on the other hand, several automatic evaluation measures have been defined over the last years. In the next three chapters, four well established automatic evaluation measures, as well as new variants and measures, will be introduced.

Nevertheless, all automatic evaluation measures are artificial values. Therefore, a good evaluation score alone does not guarantee any usability of an MT system for human users. Consequently automatic evaluation measures must be evaluated as well – in terms of their correspondence with human evaluation. Other properties of these measures can also be of interest, such as their computational complexity or the requirements regarding external resources.

1.3 Related work

From the very beginning of MT research, evaluation of its usefulness and quality was a controversial topic, in spite of its obvious necessity. An example here might be the much-discussed ALPAC report [Pierce & Carroll$^+$ 66], which caused a major throwback in MT research. In the following decades, evaluation of MT systems was carried out in several

different – but always manual – ways. Eventually, two standardization projects called EAGLES [EAG 96] took place, followed by the ISLE project. A summary of its results, as well as a brief overview of different human MT evaluation methods and standards can be found in [Popescu-Belis & Manzi$^+$ 01] and [Arnold & Balkan$^+$ 94]. Otherwise, automatic evaluation became daily practice in MT research significantly later than in other natural language processing tasks, mainly because of the difficulty of deciding on the correctness of translation, and maybe the low quality of the output of early MT systems. The Levenshtein edit distance, normalized into the Word Error Rate, was one of the first automatic evaluation measures adapted for MT. Later, evaluation measures independent of word reordering became common; for example, [Tillmann & Vogel$^+$ 97] introduced the Position independent word Error Rate. BLEU, as described by [Papineni & Roukos$^+$ 01], was the first automatic evaluation measure to become widely used as a replacement or supplement for human evaluation in evaluation campaigns and benchmarks. Later, [Doddington 02] propagated a modification of BLEU, the so-called NIST measure. Several other automatic evaluation measures have been proposed since — for example, GTM [Turian & Shen$^+$ 03], RED [Akiba & Imamura$^+$ 01], ROUGE [Lin & Och 04a], and many more. But as only the first four methods have significantly attracted attention of the research community, this work will concentrate on WER, PER, BLEU, and NIST as established measures.

1.4 Outline of this study

After an explanation of basic terms and methods in MT evaluation in Chapter 2, two well-established similarity measures, namely BLEU and the NIST measure, will be introduced in Chapter 3. The family of distance-based evaluation measures will be presented in Chapter 4. This chapter will start with the description of WER and PER as distance measures, and will then introduce the new automatic evaluation measures LJWER, INVWER, and CDER, which allow for block reordering. Preprocessing and normalization methods and related matter common to most MT evaluation measures are the subject of Chapter 5. Techniques to measure the correspondence between human and automatic evaluation will be described in Chapter 6.

After the theoretical part of this work, a practical part gives experimental evidence to the presented measures and methods. This part contains several experiments assessing the correlation of the presented evaluation measures with human evaluation, with a special regard to the different parameters, methods, and preprocessing steps described in the theoretical part. The experimental setup for this work is the topic of Chapter 7. An overview of the experimental results is given in Chapter 8. This work is concluded with a short discussion and outlook on possible further research topics in Chapter 9.

Three appendices provide additional information for the understanding and use of this work: Appendix A gives a short documentation on the MT evaluation tool that was implemented to run the experiments for this work. Appendix B contains more result tables for the experiments. Finally, Appendix C explains the notation in this work, and provides additional proofs.

Chapter 2

Basic principles of MT evaluation

In this chapter, the fundamentals for automatic evaluation measures are explained.

2.1 Different approaches to automatic evaluation

Over the last decade, several different approaches to automatic evaluation have been proposed. These evaluation methods can be differentiated on their technical specification:

2.1.1 Reference translation based measures vs. estimating measures

The majority of automatic evaluation measures for MT are based on the comparison of MT system output with a set of translations that are known to be correct, the so-called **reference translations**. Only measures of this family will be covered in this work, even though there are different approaches as well. For example, the assessment of quality scores to sentences can be seen as a classification problem, where the class (the score) of a candidate sentence is to be estimated by an automatic classificator. Examples of such a classificator are the nearest neighbor estimator, as in [Vogel & Nießen$^+$ 00], or a classificator using language-based features, as in described in [Blatz & Fitzgerald$^+$ 03].

2.1.2 Similarity measures vs. error measures

A rather obvious differentiation is the one between similarity measures and error measures. The higher a similarity score is, the better is the candidate translation in terms of the measure. The higher an error measure is, the worse is the candidate translation in terms of the measure.

Except that this has to be taken into account when comparing evaluation scores, the only practical implication is that there usually is a lower limit for error rates, whereas there is an upper limit for similarity measures, because a candidate sentence will by definition never be better in terms of translation quality than a reference sentence.

2.1.3 String-based measures vs. count-vector–based measures

In contrast to the speech recognition process, reordering of words is an integral part of the MT process, or more precisely the MT generation process. But reordering has also to be taken care of in the MT evaluation process, as the correctness of a sentence does not necessarily change significantly on a reordering of its words. Basically, there are two different approaches for an automatic evaluation measure to handle different ordering of the words in a candidate sentence:

On the one hand, the candidate sentence can explicitly be considered as ordered. The word order is then incorporated directly when calculating the measure. Reordering can also be taken into account, but have to be so explicitly. One way to do this is providing multiple reference sentence covering all admissible permutations. Another, more flexible way is defining block movement operations for the measure.

On the other hand, the evaluation measure can consider sentences to be basically unordered — for example, by comparing only their word or m-gram count vectors. If m-gram count vectors are used, the ordering of the words within the sentences will implicitly gain importance – the larger the m-gram length, the higher will be the effect of a reordering.

Hybrids of these approaches are imaginable; for example, occurrences at certain positions or regions of a sentence can be counted.

2.1.4 Different ways to handle multiple reference sentences

In automatic speech recognition, each task has only one absolutely correct outcome. In MT, there are usually more than just one correct outcomes, as there will be many ways to translate a sentence correctly. Therefore, most MT evaluation test sets contain more than one reference translation for each source sentence. Automatic evaluation measures must be able to take care of this; they can consequently be distinguished on the way they handle multiple references:

On the one hand, an evaluation measure can treat each reference sentence separately. In this approach, the measure will be calculated for the candidate sentence and each of the reference sentences separately. All reference-wise scores are then combined into a single candidate-wise score. Possible ways to combine the reference scores are taking the minimum distance, or the maximum similarity. More sophisticated combination methods could be defined, but are not in common use.

On the other hand, an evaluation measure can "pool" the reference sentences such that only this pool is compared to the candidate sentence, instead of each reference on its own.

2.2 Demands on an automatic evaluation measure

Independent of how a measure is defined and implemented, there are certain points it is required to fulfill, namely reproducibility, a tolerable computational complexity, and a high correspondence with human evaluation.

2.2.1 Reproducibility

For competitive research in the scientific community, the reproducibility of results is essential. If the ranking in a campaign could not be verified, it would be almost as worthless as an announced breakthrough in MT research that no other group can reproduce. Consequently, it is necessary that such campaigns are conducted using documented evaluation methods and measures only. Well-established measures have the additional advantage here that many research groups will already have evaluation tools available, as well as experience in their usage and properties.

Moreover, an evaluation measure should be independent of its implementation: Certain alignment problems are NP-hard, so it is tempting to define an evaluation measure that is actually based on a hard problem, and give only an approximating algorithm for it. The trouble here lies in the tendency of reimplementations of such a measure to have a different approximation with to different results.

Furthermore, demanding reproducibility has consequences to the size of the set of parameters for an evaluation measure. Many evaluation measures depend on parameters, such as substitution costs, as well as on certain preprocessing steps. Together with the set of evaluation data, such as the reference corpus, all this must be known and published for evaluation results to be reproducible. Generally, this means that the fewer parameters an evaluation measure depends on, the better is its reproducibility.

2.2.2 Reasonable complexity

MT itself is used by researchers as well as by end users in everyday use. In contrast, MT evaluation takes place only in research. Consequently, CPU and memory limits are generally much more relaxed for MT evaluation than for MT itself. Nevertheless, a long run time for the MT evaluation step can also be unwanted for, especially when it is one of many steps in MT parameter training.

Moreover, many MT development corpora consist of newspaper articles and similar texts with a sentence length of up to a hundred words, and more. For sentence lengths of this magnitude, algorithms with an exponential run time or space complexity are unsuitable. Even algorithms with a polynomial run time can be impracticable in these cases, if the polynomial has a high degree.

2.2.3 Correspondence with human evaluation

The most important criterion to keep in mind is that MT is targeted for human use. Translations generated by a MT system are usually meant to be read and understood by human users, rather than by machines. For this, automatic evaluation measures have to reflect the quality of MT with regard to human understanding. In short, there should be a high correlation between automatic and human evaluation.

Chapter 3

Similarity measures based on m-gram count vectors

Two established MT evaluation measures based on m-gram precision are presented in this chapter: BLEU and the NIST measure.

3.1 BLEU

3.1.1 Genuine BLEU

In 2000, Papineni and his team [Papineni & Roukos$^+$ 01] introduced an evaluation measure for MT they called **BiLingual Evaluation Understudy** (BLEU).

BLEU has the following properties:

- It is based on m-gram **count vectors**
- For multiple references, **pooling** takes place
- It is a **precision** measure
- The precision is modified such that each candidate m-gram is considered correct at most as many times as it occurs in at least one reference sentence
- A **brevity penalty** is added to avert a bias towards short sentences consisting of "safe guesses" only

Formally, let $n_{e^m,k}$ be the count of m-gram e^m in candidate sentence E_k. Analogously, let $\widetilde{n}_{e^m,r,k}$ be its count in reference sentence $\widetilde{E}_{r,k}$ of the candidate sentence. N_m is the total candidate m-gram count. The pooling of the reference sentences for E_k is then accomplished by the calculation of a maximum m-gram count vector $\widetilde{n}_{e^m,k}$. For each m-gram, this count vector stores the maximum number of occurrences over the reference sentences $\widetilde{E}_{r,k}$:

$$\widetilde{n}_{e^m,k} := \max_r \widetilde{n}_{e^m,r,k} \tag{3.1}$$

With the pooled maximum m-gram count from Equation (3.1), the m-gram co-occurrence count $n^{\cap}_{e^m,k}$, which is needed for the modified m-gram precision, is defined as:

$$n^{\cap}_{e^m,k} := \min\left(n_{e^m,k}, \widetilde{n}_{e^m,k}\right) \tag{3.2}$$

or, accumulated over candidate set:

$$n^{\cap}_m := \sum_k \sum_{e^m \in E_k} n^{\cap}_{e^m,k} \tag{3.3}$$

BLEU is then the w_m-weighted geometric mean of the modified m-gram precision for $m = 1, \ldots, M$, smoothed by terms s_m and multiplied by a length dependent brevity penalty lp_{BLEU}:

$$\text{BLEU} := lp_{BLEU} \exp\left\{ \sum_{m=1}^{M} w_m \log\left(\frac{n^{\cap}_m + s_m}{N_m + s_m} \right) \right\} \tag{3.4}$$

The BLEU definition leaves open the possibility to weight m-gram lengths differently. However, all actual implementations apply equal weights $w_m := \frac{1}{M}$. The **smoothing terms** s_m are zero. Studies have been made about the optimal maximum m-gram length M for BLEU. $M = 4$ is considered gold standard here.

For the length penalty, let L^*_{tot} be the total reference length, and I_{tot} be the total candidate length. The BLEU brevity penalty is then defined as:

$$lp_{BLEU} := \min\left(1, \exp\left(1 - \frac{L^*_{tot}}{I_{tot}}\right)\right) \tag{3.5}$$

3.1.2 Smoothed BLEU: BLEU-S, BLEU-S'

In the case of short evaluation corpora, and especially when regarding single sentences only, it is not unlikely for an m-gram co-occurrence count $n^{\cap}_m$ to be zero for large m. If this happens, the whole BLEU score becomes zero*. To allow for sentence-wise evaluation, [Lin & Och 04b] define the BLEU-S measure with

$$s_1 := 0 \qquad \text{and} \qquad s_{m>1} := 1. \tag{3.6}$$

Therefore, BLEU-S will not become zero unless even the unigram co-occurrence is zero. For large co-occurrence counts, the difference between BLEU and BLEU-S becomes immeasurable. But as most experiments of this study were performed on sentence level, this smoothing technique for BLEU has been adopted for this work. Additionally, experiments have been conducted with an alternative smoothing term of

$$s'_1 := 0 \qquad \text{and} \qquad s'_{m>1} := \begin{cases} 0.5 & \text{if } n^{\cap}_m = 0 \\ 0 & \text{otherwise.} \end{cases} \tag{3.7}$$

Anyhow, this approach did neither show any advantage over BLEU-S in experiments on single sentences, nor a measurable difference to BLEU and BLEU-S on system level; therefore no further research on this has been conducted.

*with the general geometric mean. In Equation (3.4), BLEU becomes undefined in this case.

3.2 The NIST score

[Doddington 02] later enhanced BLEU into the NIST measure to abolish the necessity for smoothing at sentence level evaluation, to reduce problems unwanted effects of the BLEU brevity penalty, and to take the different importance of different m-grams into account. For the latter, the NIST measure weights each m-gram by the information gain of the m-gram itself and its $(m-1)$-prefix. These NIST **information weights** are defined as:

$$\mathit{Info}(e^m) := -\left(\log_2 \widetilde{N}_{e^m} - \log_2 \widetilde{N}_{e^{m-1}}\right) \tag{3.8}$$

Notice that frequent m-grams in the reference corpus are considered to be less important by this definition. In consequence, the weight of a phrase occurring in many reference sentences for a candidate is considered to be lower than the weight of a phrase occurring only once. A discussion on the reasons and consequences of this notion would be beyond the scope of this study.

The NIST score is the sum over all information weights of the co-occurring m-grams, summed up separately for each $m = 1, \ldots, M$, and normalized by the total m-gram count:

$$\text{NIST} := \text{lp}_{\text{NIST}} \cdot \sum_m \left(\frac{1}{N_m} \cdot \sum_k \sum_{e^m \in E_k} n^{\cap}_{e^m,k} \cdot \mathit{Info}(e^m) \right) \tag{3.9}$$

The adapted brevity penalty to avoid a bias towards short candidates is defined as follows:

$$\text{lp}_{\text{NIST}} := \exp\left(\beta \cdot \log_2^2 \min\left(1, \frac{I_{tot}}{L^*_{tot}}\right)\right) \tag{3.10}$$

β is chosen such that $\text{lp}_{\text{NIST}}(I_{tot} = \frac{2}{3} L^*_{tot}) = \frac{1}{2}$; that is,

$$\beta := -\frac{\log_2 2}{\log_2^2 3} \tag{3.11}$$

3.3 Algorithms for BLEU and NIST

With hash tables or similarly efficient data structures, the count vector can be constructed in time linear in the sentence length. From the count vector, occurrence counts can be calculated in linear time as well. Both BLEU and NIST have thus a time complexity of

$$O\left(R \cdot M \cdot (I + K \cdot L)\right) \tag{3.12}$$

Otherwise, there are no algorithmic challenges in the implementation of these measures.

Chapter 4

Distance-based evaluation measures

In this chapter, two families of distance measures for the evaluation of machine translation are presented. For each family, several new approaches as well as efficient algorithms for their calculation are introduced.

4.1 Introduction

Distance-based automatic evaluation measures compare a candidate sentence with its reference sentences using a **distance function**. This distance is zero if candidate sentence and reference sentence are equal, and it is the higher the more different candidate and reference sentence are. To obtain an **error rate** ER that is comparable among sentences of different length, the absolute distance is normalized by the length of the reference sentence.

4.1.1 Distance measures for multiple reference sentences

If there are multiple reference sentences for a candidate, the distance to the nearest sentence is relevant for the score; that is, the minimum distance to all reference sentences. Alternative methods are possible — for example, taking the average distance instead of the minimum distance — but are rarely used in practise. For a whole candidate set, all distances of the candidate sentences set are summed up. Including length normalization, the calculation of the error rate ER of a whole candidate set thus looks as follows, with a distance measure d:

$$\mathrm{ER} := \frac{1}{L^*_{tot}} \sum_k \min_r d\left(E_k, \widetilde{E}_{r,k}\right) \tag{4.1}$$

Summing up the absolute distances weights the sentences implicitly by their length in the system score. Normalizing the score for each candidate sentence, and then summing up the normalized scores could avoid this weighting, although the latter scheme is rarely used in practice.

4.1.2 Distance measures and metrics

A binary function $d(x, y)$ is called a **distance measure** if it satisfies

$$\forall x, y : d(x, y) \geq 0 \qquad \textbf{(positive)} \tag{4.2}$$
$$\forall x, y : d(x, y) = 0 \Leftrightarrow x = y \qquad \textbf{(isolating)} \tag{4.3}$$
$$\forall x, y : d(x, y) = d(y, x) \qquad \textbf{(symmetric)} \tag{4.4}$$

If the measure additionally satisfies

$$\forall x, y, z : d(x, y) + d(y, z) \geq d(x, z) \qquad \textbf{(triangular)} \tag{4.5}$$

it is called a **metric***.

Another important property of a measure is whether it is *convex*; that is, whether the distance between the sum (concatenation) of two candidate and reference sentences is always lower or equal to the sum of the distance between the sentences separately. This is the case exactly if the following inequality holds:

$$\forall x, x', y, y' : d(x, y) + d(x', y') \geq d(xx', yy') \qquad \textbf{(convex)} \tag{4.6}$$

4.1.3 A simple metric: $d_{\Delta I}$

A simple distance measure is the **length difference** of the candidate sentence length I and the reference sentence length L. With cost parameters, c_{DEL} and c_{INS}— usually both set to 1 — to weight positive and negative length differences accordingly, $d_{\Delta I}$ is:

$$d_{\Delta I}(E, \widetilde{E}) := \begin{cases} (I - L) \cdot c_{DEL} & \text{if } I \geq L \\ (L - I) \cdot c_{INS} & \text{if } I < L \end{cases} \tag{4.7}$$

Although the length difference has no practical meaning as an evaluation measure by itself, it is a lower bound for all distance measures presented here. Moreover, the length difference can be calculated in constant time and space. Consequently, it can serve as a quick estimate for most distance measures.

Two sentences with the same length will always have a $d_{\Delta I}$ of zero, even if they are not equal. Therefore, the isolation axiom holds only for the identity of lengths, rather than for the identity of the words.

4.1.4 Edit operations

Most distance measures for MT can be defined using varying sets of **edit operations** Ops These operations can be word-based or block-based, depending on the distance measure they are used for. Common operations are **substitution deletion** or **insertion** of words. The edit distance approach assumes that the candidate sentence E is edited into the reference sentence $\widetilde{E}$. Editing is defined using a sequence of edit operations:

$$\mathrm{Ops}(E, \widetilde{E}) := \left\{ \mathrm{op}_1^P \;\middle|\; E \underset{\mathrm{op}_1^P}{\longrightarrow} \widetilde{E} \right\} \tag{4.8}$$

Each of these operations is assigned a cost c_{op}. This cost can be fixed, or it can depend on the edited words. The distance is then defined as total cost of the necessary edit operations. If there is more than one possible sequence of edit operations to edit the candidate sentence into the reference sentence, the sequence with minimum costs is taken:

$$d(E, \widetilde{E}) := \min_{\mathrm{op}_1^P \in \mathrm{Ops}} \sum_p c_{\mathrm{op}_p} \tag{4.9}$$

Within the usual interpretation of these edit operations, a substitution operation can always be replaced by a deletion and a following insertion operation. Therefore, for the rest of this chapter it will be assumed that the following **cost triangular inequality** holds:

$$c_{\mathrm{SUB}} \leq c_{\mathrm{INS}} + c_{\mathrm{DEL}} \qquad \textbf{(cost-triangular)} \tag{4.10}$$

or, for word dependent costs

$$\forall e, \tilde{e}: \; c_{\mathrm{SUB}}(e, \tilde{e}) \leq c_{\mathrm{DEL}}(e) + c_{\mathrm{INS}}(\tilde{e}) \qquad \textbf{(cost-triangular')} \tag{4.10a}$$

4.2 Count-vector–based distance measures

The first family of distance measures for MT evaluation presented here is based on **count vectors**: For each word or m-gram, its number of occurrences in candidate and reference sentence is counted; the counts are stored in a candidate and a reference count vector. These count vectors are then compared. In contrast to the BLEU or NIST scheme, separate count vectors and thus separate distances are calculated for each reference sentence. As with other distance measures, the lowest count vector distance is then selected.

4.2.1 PER

The **Position independent Error Rate** (PER) [Tillmann & Vogel$^+$ 97] is based on the idea that the candidate sentence is edited into the reference sentence without any regard for the word order. Valid edit operations are *insertion*, *deletion*, and *substitution* of single words. These position independent edit operations can easily be implemented as operations on the count vectors of candidate and reference sentence. An *insertion* operation means in this context that the count of a specific word is increased by one in the count vector. Analogously, a *deletion* corresponds to a decrease by one. A *substitution* means that the count of one word is increased, and the count of another word is decreased in exchange.

PER for fixed costs

Given the count vectors n_e, $\widetilde{n}_e$, and arbitrary but fixed costs satisfying Equation (4.10), d_{PER} can be calculated as

$$d_{PER}(E,\widetilde{E}) := \frac{1}{2}\Big(\sum_e |n_e - \widetilde{n}_e| - |I-L|\Big)\cdot c_{SUB} + \begin{cases} |I-L|\cdot c_{DEL} & \text{if } I \geq L \\ |I-L|\cdot c_{INS} & \text{otherwise} \end{cases} \tag{4.11}$$

If $c_{INS} = c_{DEL} = c_{SUB} = 1$, this can be simplified to

$$d_{PER}(E,\widetilde{E}) := \frac{1}{2}\Big(\sum_e |n_e - \widetilde{n}_e| + |I-L|\Big) \tag{4.12}$$

From another point of view, the position independent edit distance d_{PER} can be regarded as **Earth Mover's Distance** [Rubner & Tomasi$^+$ 98], with the candidate counts as suppliers, and the reference counts as consumers. If candidate sentence and reference sentence have different lengths, **empty words** ε must be added to balance the count vectors. Moves to and from these empty words correspond to insertions and deletions. Moves among non–empty words correspond to substitutions and zero cost identity operations.

PER for word dependent costs

If the costs of the edit operations are not fixed, but dependent on the words that are edited, Equation (4.11) does not hold. Instead, d_{PER} can be calculated as the cost of a solution to the following assignment problem: Each candidate word and each reference word is a node. Edges go from all candidate words to all reference words. Edge costs are the substitution costs between the node words, or zero if the words are identical. If either sentence is shorter than the other, empty words ε prolong the shorter one such that both sentences have the same length. The edge costs from or to these empty words are the deletion or insertion costs respectively.

Figure 4.1 shows an example of such an alignment graph for the sentences `we have been there` and `we were there`. Identity edges with zero costs connect each of the word pairs `we/we` and `there/there`. Substitution edges connect all other combinations `we/were`, etc., on corresponding substitution costs. Additionally, one empty word node ε had to be added to the second sentence. Deletion edges connect ε with each word of the first sentence. An optimal alignment for this problem is then `we/we`, `have/were`, `been/`ε , `there/there`.

The **Hungarian Algorithm** can solve this assignment problem – finding a minimum cost matching in a bipartite graph – in time $O\big(\min(I^2L, L^2I)\big)$. [Knuth 93] gives a detailed description and implementation of this algorithm.

Multiset distance: MSDER

A similar count vector distance is the **multiset distance**. This distance is the sum of the absolute differences in the count vectors. As it can basically be regarded as PER without substitutions, it can be defined and calculated as PER with costs of $c_{INS} = c_{DEL} = 1$ and $c_{SUB} = c_{INS} + c_{DEL} = 2$.

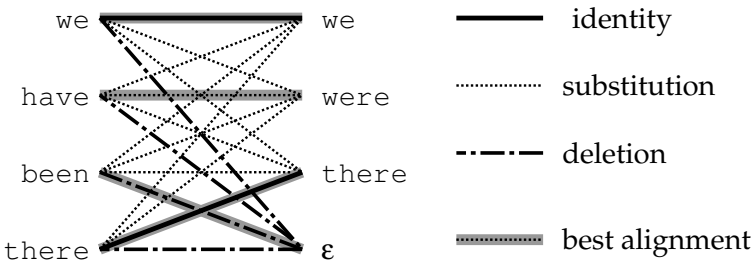


Figure 4.1: Example of a PER alignment graph.

4.2.2 m-PER

The definition of the original PER distance rests upon unigram count vectors. To enhance PER, it is a simple step to extend the distance to bigrams or arbitrary m-grams by calculating the distance on bigram or arbitrary m-gram count vectors. Reordering of substrings in a sentence will cause an m-gram mismatch at the substring boundaries. Consequently, the ordering of the words in the sentence gains importance in comparison with its complete neglect in the original PER.

An evaluation measure can combine m-PER distances for different m into one distance by summing up over the absolute or normalized distances. For multiple references, the summing up can take place either before the minimization in Equation (4.1), or afterwards. In the former case, each candidate sentence has exactly one nearest reference sentence with regard to m-PER. In the latter case, it can have different nearest candidate sentence for each m.

4.2.3 Skip bigram PER, ROUGE-S

[Lin & Och 04a] introduce an evaluation measure called **Recall-Oriented Understudy for Gisting Evaluation by Skip bigrams** (ROUGE-S). Basically, the ROUGE-S measure compares **skip bigram** count vectors. *skip bigrams* are pairs of arbitrary words from a sentence appearing in the same order in both sentences. The **skip** (i.e., the number of words between the skip bigram parts in the sentence) can be limited. For a **maximum skip** S, this reduces the worst case size of the count vector from $O\big((I+L)^2\big)$ to $O\big(S \cdot (I+L)\big)$. The lower the maximum skip is, the more "local" is the emphasis of the resulting skip bigram measure. The original ROUGE-S measure is implemented as an **F-measure** (i.e., a combined *precision* and *recall* measure) on skip bigrams.

Sentence:	`I prefer the plane`
skip bigrams with skip=0:	`[I -- prefer], [prefer -- the], [the -- plane]`
skip bigrams with skip=1:	`[I -- the], [prefer -- plane]`
skip bigrams with skip=2:	`[I -- plane]`

Figure 4.2: Example of skip bigrams.

The idea of comparing *skip bigram* count vectors can be transferred to the PER distance. *skip bigrams* can be inserted to, deleted from, and substituted in the count vector; the *skip bigram* distance is then the minimum edit cost to transform the candidate *skip bigrams* into the reference *skip bigrams*. An appropriate normalization constant here is the total reference *skip bigram* count.

4.3 A string-based distance measure: Levenshtein distance, WER

4.3.1 Definition and Levenshtein alignment grid

The **Levenshtein distance** d_{LEV} is the "classical" edit distance for strings. As described in [Levenshtein 66], candidate and reference sentence are treated in order. Deletions, insertions, and substitutions of words are the only admitted edit operations. The **Word Error Rate** WER is then the error rate induced by the Levenshtein distance.

Finding the optimal set of Levenshtein operations for Equation (4.9) can be reduced to finding the cheapest path in a **Levenshtein alignment grid** as shown in Figure 4.3. Nodes correspond to pairs of positions between words in candidate and reference sentence. Horizontal edges correspond to deletion operations – a candidate word is passed, but no reference word – and have an edge cost of c_{DEL}. Vertical edges correspond to insertion operations, with an edge cost of c_{INS}. Depending on whether the words corresponding to the edge are equal, diagonal edges correspond to identity operations, with zero edge costs, or substitution operations, with edge costs of c_{SUB}. The Levenshtein distance between candidate and reference sentence is then equal to the cost of the cheapest path between the start node and the end node of the alignment grid. Here, the start node is the lower left node in the graph, before the first words of both sentences. The end node is the upper right node, after the last words of both sentences.

4.3.2 A dynamic programming approach

For an efficient calculation of the Levenshtein distance using the **dynamic programming** (DP) approach [Cormen & Leiserson$^+$ 90, Levenshtein 66], the following **auxiliary quantity** $Q(i,l)$ is useful:

$$Q(i,l) := d_{LEV}\left(e_1^i, \tilde{e}_1^l\right) \tag{4.13}$$

For Q, the following recursion holds:

$$\begin{aligned} Q(0,0) &= 0 \\ Q(i,l) &= \min\left\{ \begin{array}{l} Q(i-1,l-1) + c_{SUB}(e_i,\tilde{e}_l)\cdot\left(1-\delta(e_i,\tilde{e}_l)\right), \\ Q(i-1,l) + c_{DEL}(e_i), \\ Q(i,l-1) + c_{INS}(\tilde{e}_l) \end{array} \right\} \end{aligned} \tag{4.14}$$

Algorithm 4.1 solves this recursion in time $O(I \cdot L)$. DP can be applied because the recursive calculating of $Q(i,l)$ requires values $Q(i',l')$ for indices of $i' \leq i$ and $l' \leq l$ only.

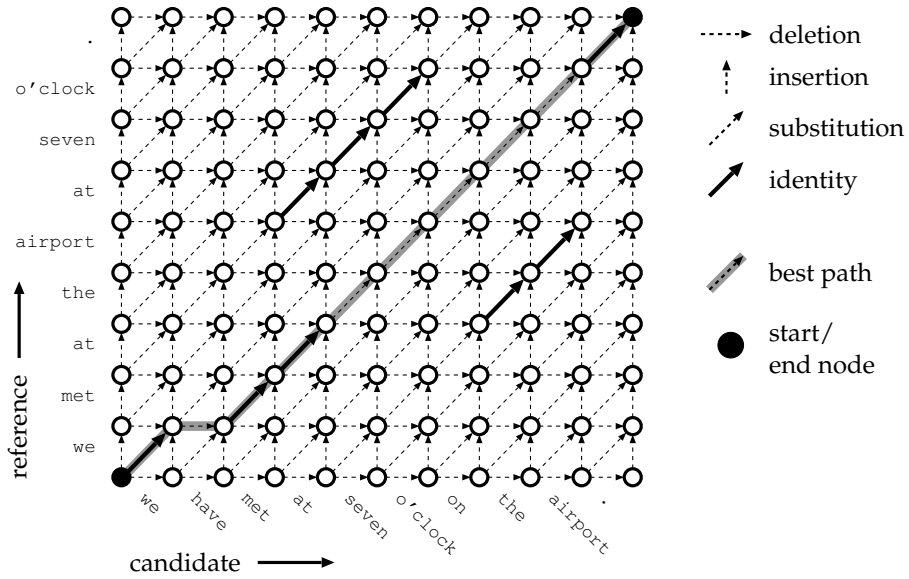


Figure 4.3: Example of a Levenshtein alignment grid with optimal path.

Algorithm 4.1 Dynamic programming algorithm for the Levenshtein distance.
From [Levenshtein 66].

for $i := 0$ **to** I **do**
  $Q(i, 0) \leftarrow i \cdot c_{\mathrm{DEL}}(e_i)$
end for
for $l := 1$ **to** L **do**
  $Q(0, l) := Q(0, l-1) + c_{\mathrm{INS}}(\tilde{e}_l)$
  for $i := 1$ **to** I **do**

$$Q(i, l) := \min \left\{ \begin{array}{l} Q(i-1, l-1) + c_{\mathrm{SUB}}(e_i, \tilde{e}_l) \cdot \big(1 - \delta(e_i, \tilde{e}_l)\big), \\ \quad Q(i-1, l) + c_{\mathrm{DEL}}(e_i), \\ \quad Q(i, l-1) + c_{\mathrm{INS}}(\tilde{e}_l) \end{array} \right\}$$

  end for
end for

4.4 Block movements for string-based distance measures

The Levenshtein distance treats words only in strict order. Reordering of words or phrases between the sentences can be modelled only using a chain of insertions, substitution, and deletions. The costs for this is proportional to the length of the reordered phrases. This is a disadvantage for the application in MT evaluation because the reordering of words and phrases within a sentence is a frequent phenomenon in MT. To circumvent this strictness by allowing for reordering while still maintaining the local order of the words, **block movements** can be introduced: In this approach, candidate and reference sentences are assumed to consist of parallel word blocks. Within each such block, candidate and reference words are Levenshtein-aligned. Blocks may be distributed arbitrarily over the sentence, with the additional constraint that the blocks form a **complete** (C) and **disjunct** (D) coverage of both the candidate sentence and the reference sentence.

If there is no penalty for too large numbers of blocks, arbitrary reordering of single words is possible. In this case the block movement distance is equal to d_{PER}. Consequently, such a measure should penalize block alignments by a **block cost** c_{BLOCK} for each separate nonconsecutive blocks in the alignment.

4.4.1 Long jump distance, LJWER

As the number of such blocks is equal to the number of gaps among the blocks plus one, the block costs can equivalently be expressed using a **long jump** operation that jumps over the gaps between each two blocks. In the alignment grid, long jump operations correspond to edges between arbitrary nonconsecutive nodes from the same row. Figure 4.4 gives an example of such a long jump grid, along with its cheapest alignment path.

The **long jump distance** d_{LJ}, or CDCD-distance, is then defined as the cost of the cheapest path from the first node to the last node in the long jump alignment grid. The path must hold the additional constraint that the Levenshtein edges along the path must cover each candidate and reference word exactly once. For reasonable fixed costs, Equations 4.2 to 4.6 hold. Consequently, d_{LJ} is a *convex metric*.

The difficulty arising for the calculation of this distance is that this is an NP-hard problem. A proof can be found in [Lopresti & Tomkins 97]. For the present study, an adaption of the **Held–Karp Algorithm** for word reordering [Held & Karp 62, Tillmann & Ney 00] has been implemented. However, the exponential run time of this algorithm inhibited any experiments for sentence lengths exceeding about twenty words.

4.4.2 $\overline{\mathrm{CD}}$CD-distance, CDER

For the long jump distance, an alignment path must form a *complete* (C) and *disjunct* (D) coverage of both candidate sentence and reference sentence. It is nevertheless imaginable to drop one or both of these constraints for candidate or reference sentence. The search of the optimal path must then be adapted to account for the larger search space. As [Lopresti & Tomkins 97] show, the block distance problem becomes polynomial if both C and D are dropped for at least one of candidate and reference sentence. For the following,

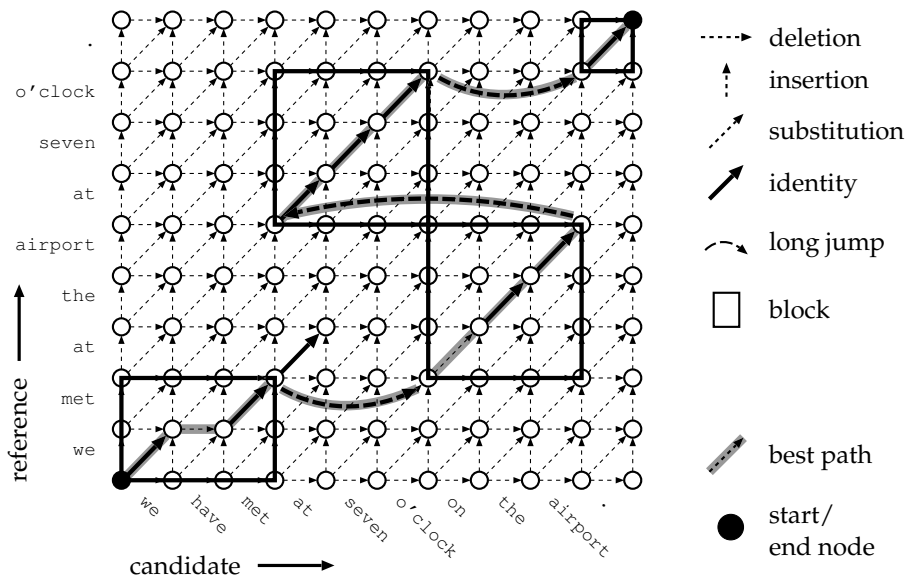


Figure 4.4: Example of a long jump alignment grid. For clarity, only long jump edges from the best path are drawn.

let C and D hold for the reference sentence $\widetilde{E}$ only.[†] Every candidate word is instead allowed to be covered arbitrarily often; that is, once, several times, or no time at all. Then, define $d_{\overline{CD}CD}$ to be the long jump distance d_{LJ} under these relaxed constraints. In the notation of [Lopresti & Tomkins 97], this is the **$\overline{\text{CD}}$CD distance** between the candidate sentence and the reference sentence. Again, an *auxiliary quantity* $Q(i,l)$ for use in dynamic programming can be defined as

$$Q(i,l) := d_{\overline{CD}CD}\left(e_1^i, \tilde{e}_1^l\right) \tag{4.15}$$

For this auxiliary quantity Q, the following modification of the Levenshtein recursion holds:

$$\begin{aligned} Q(0,0) &= 0 \\ Q(i,l) &= \min\left\{ \begin{array}{l} Q(i-1,l-1)+c_{SUB}(e_i,\tilde{e}_l)\cdot\delta(e_i,\tilde{e}_l), \\ \quad Q(i-1,l)+c_{DEL}(e_i), \\ \quad Q(i,l-1)+c_{INS}(\tilde{e}_l), \\ \min\limits_{i'} Q(i',l)+c_{LJ} \end{array} \right\} \end{aligned} \tag{4.16}$$

Whereas [Lopresti & Tomkins 97] gives an implementation with $O(I^2 \cdot L)$ time complexity, this recursion can be solved in linear time $O(I \cdot L)$ using a modification of the Levenshtein DP algorithm (4.1), as shown in Algorithm 4.2 .

In MT, candidate translations tend to contain omissions of some parts and repetitions of other parts from reference translations. These omissions and repetitions are wrong in

[†] An explanation for this choice is given in the following paragraphs.

Algorithm 4.2 Dynamic programming algorithm for the $\overline{\text{CD}}$CD distance.

for $i := 0$ **to** I **do**
  $Q(i, 0) \leftarrow \min\{c_{LJ}, i \cdot c_{DEL}(e_i)\}$
end for
for $l := 1$ **to** L **do**
  $Q(0, l) \leftarrow Q(0, l-1) + c_{INS}(\tilde{e}_l)$
  for $i := 1$ **to** I **do**

$$Q(i, l) \leftarrow \min\left\{\begin{array}{l} Q(i-1, l-1) + c_{SUB}(e_i, \tilde{e}_l) \cdot \delta(e_i, \tilde{e}_l), \\ Q(i-1, l) + c_{DEL}(e_i), \\ Q(i, l-1) + c_{INS}(\tilde{e}_l) \end{array}\right\}$$

  end for
  $q_{rowmin} \leftarrow \min_{i'} Q(i', l)$
  for $i := 1$ **to** I **do**

$$Q(i, l) \leftarrow \min\left\{\begin{array}{r} Q(i, l), \\ q_{rowmin} + c_{LJ} \end{array}\right\}$$

  end for
end for

most cases. Reference translations, on the other hand, will rarely contain wrong repetitions, and are expected to be complete. This observation is why a complete and disjunct coverage is demanded of the reference sentence instead of the candidate sentence: Assume that a candidate sentence contains "forbidden" repetitions of blocks; that is, repetitions that do not occur literally in the reference sentence. Then, these repetitions cannot be detected in either direction of the CD constraint, except that each repeated block requires additional long jumps with costs independent of the block length. However, omissions in the candidate sentence can be detected only if a complete coverage of the reference sentence is demanded. Consequently, $d_{\overline{\text{CD}}\text{CD}}$ can be seen as a **recall**-oriented measure, because deletions – words that can be found only in the reference sentence – are penalized, whereas insertions – words that can be found only in the candidate sentence – are not, except for the constant long jump cost. The correlation experiments in Section 8.9 confirm the supposition that requiring CD for the reference sentence is the reasonable direction for $d_{\overline{\text{CD}}\text{CD}}$ in MT evaluation.

4.4.3 $\overline{\text{CD}}$CD-distance with miscoverage penalty

Because of the relaxed constraints for the admissible alignment paths, $d_{\overline{\text{CD}}\text{CD}}$ is a lower bound for the long jump distance d_{LJ}. In a $d_{\overline{\text{CD}}\text{CD}}$ alignment grid, candidate words can have a coverage different from one, whereas the coverage of all candidate and reference words in a d_{LJ} alignment grid with a valid path is always one. Consequently, the miscoverage is an indicator of how much d_{LJ} and $d_{\overline{\text{CD}}\text{CD}}$ differ for a given sentence pair E, $\widetilde{E}$. With an appropriate miscoverage penalty, $d_{\overline{\text{CD}}\text{CD}}$ plus this penalty could serve as an approximation of d_{LJ}. Two definitions of such a penalty have been studied for this work:

Length difference

Assume $c_{LJ}, c_{DEL} = const$. Provided that $c_{LJ} \leq c_{DEL}$, there is always an optimal $\overline{CD}CD$ alignment path that does not contain any deletion edges. Therefore, if the candidate sentence is longer than the reference sentence by $I - L$ words, at least $I - L$ deletion edges will be found in any d_{LJ} path. The described optimal $\overline{CD}CD$ path, on the other hand, will not contain any deletion edge. Consequently, the *length difference* makes a useful miscoverage penalty:

$$lp_{\Delta L} := \max\big(I - L, 0\big) \cdot c_{DEL} \tag{4.17}$$

As this penalty is independent of the $d_{\overline{CD}CD}$ alignment path, the search algorithm does not have to take care of the penalty in the optimization.

Absolute miscoverage

The attachment of deletion edges for uncovered candidate words, and of insertion edges for over-covered candidate word gives a method to construct a valid – but not necessarily optimal – d_{LJ} path out of a $d_{\overline{CD}CD}$ path. Figure 4.5 illustrates this procedure. Accordingly, the **absolute miscoverage** can be used as a miscoverage penalty lp_{misc} for $d_{\overline{CD}CD}$:

$$lp_{misc} := \sum_i \begin{cases} c_{DEL} & coverage(i) = 0 \\ 0 & coverage(i) = 1 \\ c_{INS} \cdot \big(coverage(i) - 1\big) & coverage(i) \geq 2 \end{cases} \tag{4.18}$$

where $coverage(i)$ is the number of times a substitution, insertion, or identity edge in the best alignment path visits e_i. This miscoverage penalty is not independent of the alignment path. Consequently, Algorithm 4.2 will not necessarily find an optimal solution for the sum of $d_{\overline{CD}CD}$ and lp_{misc}.[‡] With these miscoverage penalties, cheap lower and upper bounds for d_{LJ} can be calculated, because the following inequality holds:

$$d_{\overline{CD}CD}(E, \widetilde{E}) + lp_{\Delta L} \;\leq\; d_{LJ}(E, \widetilde{E}) \;\leq\; d_{\overline{CD}CD}(E, \widetilde{E}) + lp_{misc} \tag{4.19}$$

4.4.4 Inversion distance, INVWER

A possible way to achieve a polynomial calculation time while guaranteeing a disjunct coverage of both the candidate sentence and the reference sentence is to restrict the structure of possible block reordering. [Wu 95] introduced the **Bracketing Transduction Grammar** framework, which is a suitable framework for a restriction to the set of possible permutations.

[‡]otherwise d_{LJ} could be calculated in polynomial time.

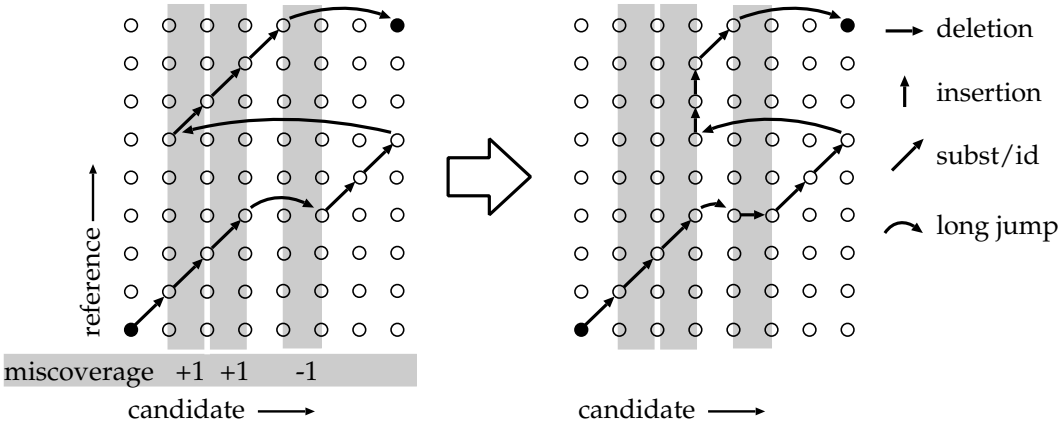


Figure 4.5: Transformation of a $d_{\overline{CD}CD}$ path into a d_{LJ} path.

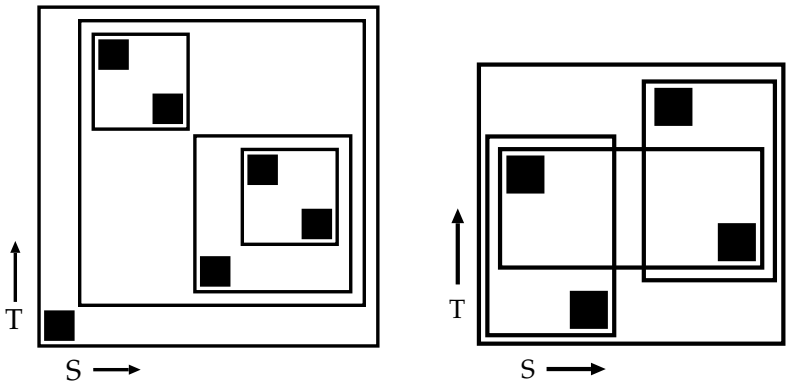
Bracketing transduction grammars

According to [Wu 95], a *Bracketing Transduction Grammar* (BTG) is a pair-of-string model that generates two output strings, S and T. It consists of one common set of production rules for both output strings. A BTG always generates a pair of sentences. Terminals are pairs of words, where each may be the *empty word* ε.

Concatenation of the terminals and nonterminals on the right hand side of a production rule is either **straight**, denoted by [AA], or **inverted**, denoted by $\langle AA \rangle$. In the former case, the parse subtree is to be read left-to-right in both S and T, and in the latter case it is to be read left-to-right in S and right-to-left in T. A BTG contains only the start symbol S and one distinct nonterminal symbol A, and each production rule consists of either a string of As or a terminal pair.

Within BTGs, all permutations must be realized as a chain of consecutive swaps of adjacent blocks. These swaps or **inversion operations** can be nested, as shown in Figure 4.6(a). The only restriction to a chain of swaps is that the boundaries of any two nested blocks must not overlap. Figure 4.6(b) gives an example of such an inadmissible overlap. Permutations that can be realized using only such nonoverlapping swaps, as the permutation in Figure 4.6(b), can thus not be described within the BTG scheme.

For the definition of a block-inversion–enabled string distance measure using the BTG scheme, all Levenshtein operations can be defined as BTG production rules, as described in [Leusch & Ueffing$^+$ 03]. To allow for reordering, an additional BTG production rule defines the **inversion** operation. This operation effects the swap of two adjacent text blocks, within the restrictions of the BTG framework. Attaching the Levenshtein operation costs c_{SUB}, c_{INS}, c_{DEL} and the inversion cost c_{INV} to the application of the corresponding production rule gives a definition of these operations in the BTG framework that is equivalent to the traditional definition. The complete list of rules is given in Table 4.1. The inversion distance d_{inv} between a candidate sentence E and a reference sentence $\widetilde{E}$ is then defined as the minimum cost of all parse trees generated by the BTG



(a) allowed by the BTG scheme

(b) forbidden by the BTG scheme

Figure 4.6: Example of nested inversions. Square dots denote aligned words in both sentences; rectangles denote blocks.

that produce this sentence pair:

$$d_{inv}(E, \widetilde{E}) := \min_{\tau \in T(E,\widetilde{E})} c(\tau) \tag{4.20}$$

Here, $T(E, \widetilde{E})$ is the set of all possible parse trees for E and $\widetilde{E}$. Notice that the minimum production cost of that BTG minus the inversion rule would be equal to the Levenshtein distance.

Example

Consider the sentence pair
`we will meet at noon in the lobby /`
`we will meet in the lobby at twelve o'clock.`

Table 4.1: Edit operations as BTG production rules.

1.	Concatenation:	$A \rightarrow [AA]$	with $c([\alpha\beta]) = c(\alpha) + c(\beta)$
2.	Inversion:	$A \rightarrow \langle AA \rangle$	with $c(\langle\alpha\beta\rangle) = c(\alpha) + c(\beta) + c_{INV}$
3.	Identity:	$A \rightarrow e/e$	with $c(e/e) = 0$
4.	Substitution:	$A \rightarrow e/\tilde{e}$, where $e \neq \tilde{e}$	with $c(e/\tilde{e}) = c_{SUB}(e, \tilde{e})$
5.	Deletion:	$A \rightarrow e/\varepsilon$	with $c(e/\varepsilon) = c_{DEL}(e)$
6.	Insertion:	$A \rightarrow \varepsilon/\tilde{e}$	with $c(\varepsilon/\tilde{e}) = c_{INS}(\tilde{e})$
7.	Start:	$S \rightarrow A$; $S \rightarrow \varepsilon/\varepsilon$	with $c(\varepsilon/\varepsilon) = 0$

Then, $d_{inv} = 3$, as these sentences can be parsed as follows (trivial concatenation brackets omitted):

$$\Big[\ \texttt{we/we will/will meet/meet}\ \Big\langle\ \ [\,\texttt{at/at}\ \underset{\sim}{\texttt{noon/twelve}}\ \underline{\underline{\epsilon/\texttt{o'clock}}}\,] \\ [\,\texttt{in/in the/the lobby/lobby}\,]\ \ \Big\rangle\ \Big]$$

The insertion rule ($\underline{\underline{\ }}$), the substitution rule ($\underset{\sim}{\ }$), and the inversion rule ($\langle\rangle$) are each applied once. The Levenshtein distance between these sentences would be 5.

Properties

If all costs are set to 1, d_{inv} is a distance measure: Concatenation and identity have zero costs; all other operation have positive costs. Therefore, d_{inv} is both *positive* and *isolating*. Moreover, d_{inv} is *symmetric*, because all production rules and costs are symmetric. Finally, the *convexity* of d_{inv} follows immediately from the concatenation rule. However, the *triangular inequality* Equation (4.5) does not hold, as a counter example proves: $d_{inv}(\texttt{abcd}, \texttt{abdc}) = 1$ and $d_{inv}(\texttt{abdc}, \texttt{bdac}) = 1$, but $d_{inv}(\texttt{abcd}, \texttt{bdac}) = 4 > 2$. Consequently, d_{inv} is not a metric.

Towards a DP recursion

Again, the dynamic programming approach can be applied to calculate d_{inv} efficiently. To obtain a suitable recursion, a closer look on d_{inv} for arbitrary substrings $e_{i_0}^{i_1}$ and $\tilde{e}_{l_0}^{l_1}$ is helpful: For the calculation of this distance, the cost of the cheapest parse tree in all parse trees $T(e_{i_0}^{i_1}, \tilde{e}_{l_0}^{l_1})$ generating these sequences has to be computed. Assuming constant costs, the following cases can be distinguished depending on the values of i_0, i_1, l_0, and l_1:

- If $i_0 = i_1$ and $l_0 = l_1$, both $e_{i_0}^{i_1}$ and $\tilde{e}_{l_0}^{l_1}$ are single words. Therefore either the identity production or the substitution production will be applied. Consequently, $d_{inv}(e_{i_0}^{i_1}, \tilde{e}_{l_0}^{l_1})$ is 0 or c_{SUB}, respectively.
- If $i_1 < i_0$, $\quad e_{i_0}^{i_1} = \epsilon$, and $\tilde{e}_{l_0}^{l_1}$ can be generated only by $l_1 - l_0 + 1$ applications of the concatenation and the insertion rule. Consequently, $d_{inv}(\epsilon, \tilde{e}_{l_0}^{l_1}) = (l_1 - l_0 + 1) \cdot c_{INS}$.
- Analogously, if $l_1 < l_0$, the deletion rule has to be applied $i_1 - i_0 + 1$ times, thus $d_{inv}(e_{i_0}^{i_1}, \epsilon) = (i_1 - i_0 + 1) \cdot c_{DEL}$.
- In all other cases, either the concatenation or the inversion production rule will be applied, hence the tree's cost include the sum of the costs of two subtrees. For the straight concatenation of blocks, it holds that

$$d_{inv}(e_{i_0}^{i_1}, \tilde{e}_{l_0}^{l_1}) = \min_{i', l'} \min_{\substack{\tau \in T(e_{i_0}^{i'}, \tilde{e}_{l_0}^{l'}) \\ \tau' \in T(e_{i'+1}^{i_1}, \tilde{e}_{l'+1}^{l_1})}} \big(c(\tau) + c(\tau')\big) \tag{4.21}$$

and for their inversion, it holds that

$$d_{inv}(e_{i_0}^{i_1}, \tilde{e}_{l_0}^{l_1}) = \min_{i',l'} \min_{\substack{\tau \in T(e_{i_0}^{i'}, \tilde{e}_{l'+1}^{l_1}) \\ \tau' \in T(e_{i'+1}^{i_1}, \tilde{e}_{l_0}^{l'})}} \left(c(\tau) + c(\tau') + c_{INV} \right) \tag{4.22}$$

The inversion distance of the substrings can then serve as *auxiliary quantity* Q for dynamic programming:

$$Q(i_0, i_1; l_0, l_1) := d_{inv}(e_{i_0}^{i_1}, \tilde{e}_{l_0}^{l_1}) \tag{4.23}$$

Then, the following dynamic programming recursion holds:

$$Q(i_0, i_1; l_0, l_1) =$$
$$\begin{cases} (l_1 - l_0 + 1) \cdot c_{INS} & \text{if } i_1 < i_0 \\ (i_1 - i_0 + 1) \cdot c_{DEL} & \text{if } l_1 < l_0 \\ (1 - \delta(e_{i_0}, \tilde{e}_{l_0})) \cdot c_{SUB} & \text{if } (i_1 = i_0) \wedge (l_1 = l_0) \\ \min\limits_{\substack{i_0 \le i' \le i_1 \\ l_0 \le l' \le l_1}} \left\{ \begin{array}{l} Q(i_0, i'; l_0, l') + Q(i'+1, i_1; l'+1, l_1), \\ Q(i_0, i'; l'+1, l_1) + Q(i'+1, i_1; l_0, l') + c_{INV} \end{array} \right\} & \text{otherwise} \end{cases} \tag{4.24}$$

Finally,

$$d_{inv}(E, \widetilde{E}) = Q(1, I; 1, L) \tag{4.25}$$

A closer look at this equation reveals that for the recursive calculation of $Q(i_0, i_1; l_0, l_1)$, values of Q only for shorter or equal substring lengths are required. Therefore, Q can be calculated using dynamic programming for increasing substring lengths $\Delta i = i_1 - i_0$ and $\Delta l = l_1 - l_0$. Calculation starts with zero length for the candidate substring and then with zero length for the reference substring. This step covers all insertions and deletions. Then, all Q values for substring lengths of 1 each are calculated. These substrings correspond to single substitution or identity operations. After that, all other Q values can be computed. A dynamic programming algorithm using this recursion can be found in Algorithm 4.3. This algorithm can be regarded as two dimensional BTG extension of the CYK algorithm from [Younger 67]. As the BTG here contains only one nonterminal A, the additional dimension for the nonterminal symbols present in the original CYK algorithm can be omitted.

From the number of nested **for**-loops in Algorithm 4.3 follows that the time complexity of this algorithm is $O(I^3 L^3)$. This complexity is noticeably higher than that of d_{LEV} or $d_{\overline{CDCD}}$, although still polynomial. The space complexity of this algorithm is $O(I^2 L^2)$.

A memoization algorithm for with pruning

Experiments indicate that in most cases it is not necessary to calculate all values of $Q(i_0, i'; l_0, l')$. Usually, it is sufficient to estimate only certain substring distances. For this, the following inequality is helpful:

Algorithm 4.3 Dynamic programming algorithm for the inversion distance.

/ All deletions */*

for $1 \leq i_0, i_1 \leq I$ **do**
  for $l \leftarrow 2$ **to** L **do**
    $Q(i_0, i_1; l, l-1) \leftarrow (i_1 - i_0 + 1) \cdot c_{DEL}$
  end for
end for

/ All insertions */*

for $i \leftarrow 2$ **to** I **do**
  for $1 \leq l_0, l_1 \leq L$ **do**
    $Q(i, i-1; l_0, l_1) \leftarrow (l_1 - l_0 + 1) \cdot c_{INS}$
  end for
end for

/ All substitutions/identities */*

for $i \leftarrow 1$ **to** I **do**
  for $l \leftarrow 1$ **to** L **do**
    $Q(i, i; l, l) \leftarrow (1 - \delta(e_i, \tilde{e}_l)) \cdot c_{SUB}$
  end for
end for

/ Concatenations/Inversions */*

for $\Delta i \leftarrow 1$ **to** $I - 1$ **do**
  for $\Delta l \leftarrow 1$ **to** $L - 1$ **do**
    for $i_0 \leftarrow 1$ **to** $I - \Delta i$ **do**
      for $l_0 \leftarrow 1$ **to** $L - \Delta l$ **do**
        $i_1 \leftarrow i_0 + \Delta i$
        $l_1 \leftarrow l_0 + \Delta l$;
        $q_{min} \leftarrow \infty$
        for $i' \leftarrow i_0$ **to** i_1 **do**
          for $l' \leftarrow l_0$ **to** l_1 **do**

$$q_{min} \leftarrow \min \left\{ \begin{array}{l} q_{min}, \\ Q(i_0, i'; l_0, l') + Q(i'+1, i_1; l'+1, l_1), \\ Q(i_0, i'; l'+1, l_1) + Q(i'+1, i_1; l_0, l') + c_{INV} \end{array} \right\}$$

          end for
        end for
        $Q(i_0, i_1; l_0, l_1) \leftarrow q_{min}$
      end for
    end for
  end for
end for
return $Q(1, I; 1, L)$

$$d_{\Delta I}(e_{i_0}^{i_1}, \tilde{e}_{l_0}^{l_1}) \leq d_{PER}(e_{i_0}^{i_1}, \tilde{e}_{l_0}^{l_1}) \leq d_{inv}(e_{i_0}^{i_1}, \tilde{e}_{l_0}^{l_1}) \leq d_{LEV}(e_{i_0}^{i_1}, \tilde{e}_{l_0}^{l_1}) \tag{4.26}$$

Therefore, both $d_{\Delta I}$ (with a time complexity of $O(1)$) and d_{PER} (with a time complexity of $O(I + L)$) provide lower bounds to d_{inv}. In addition, d_{LEV} provides an upper bound with time complexity $O(I \cdot L)$. Furthermore, for each of $d_{\Delta I}$, d_{PER}, and d_{inv}, the triangular inequality (4.5) and the convexity inequality (4.6) hold. Accordingly,strict bounds for each $Q(i_0, i'; l_0, l')$ can be estimated cheaply. Under certain conditions, these estimates can render more precise (and more expensive) calculation unnecessary, as branches in the search tree that are proven not to lead to an optimal solution can be **pruned** without loss of correctness.

Thus, let

- $lb(e_{i_0}^{i_1}, \tilde{e}_{l_0}^{l_1}) \leq d_{inv}(e_{i_0}^{i_1}, \tilde{e}_{l_0}^{l_1})$ be an arbitrary lower bound function for d_{inv} (e.g., $lb = d_{\Delta I}$, or $lb = d_{PER}$)
- lb' be another lower bound function with $lb \leq lb'$
- $ub(e_{i_0}^{i_1}, \tilde{e}_{l_0}^{l_1}) \geq d_{inv}(e_{i_0}^{i_1}, \tilde{e}_{l_0}^{l_1})$ be an arbitrary upper bound function for d_{inv} (e.g., $lb = d_{LEV}$, or $lb = d_{inv}$)
- $e_{i_0}^{i_1} / \tilde{e}_{l_0}^{l_1}$ be an arbitrary node of an optimal parse tree for d_{inv}, with $i_1 > i_0$ and $l_1 \geq l_0$ or $i_1 \geq i_0$ and $l_1 > l_0$.

Moreover, let i', l' be such that

$$e_{i_0}^{i_1} / \tilde{e}_{l_0}^{l_1} \mapsto \left[e_{i_0}^{i'} / \tilde{e}_{l_0}^{l'} \;\; e_{i'+1}^{i_1} / \tilde{e}_{l'+1}^{l_1} \right] \qquad \text{case (a)} \tag{4.27}$$

or

$$e_{i_0}^{i_1} / \tilde{e}_{l_0}^{l_1} \mapsto \left\langle e_{i_0}^{i'} / \tilde{e}_{l'+1}^{l_1} \;\; e_{i'+1}^{i_1} / \tilde{e}_{l_0}^{l'} \right\rangle \qquad \text{case (b)} \tag{4.28}$$

Then, with the abbreviation that

$$ub'(i_0, i_1; l_0, l_1) := ub(e_1^I, \tilde{e}_1^L) - lb(e_1^{i_0-1} e_{i_1+1}^I, \tilde{e}_1^{l_0-1} \tilde{e}_{l_1+1}^L) \tag{4.29}$$

the following inequalities hold:

$$ub'(i_0, i_1; l_0, l_1) \geq lb'(e_{i_0}^{i_1}, \tilde{e}_{l_0}^{l_1}) \tag{4.30}$$

and

$$ub'(i_0, i_1; l_0, l_1) \geq \begin{cases} lb'(e_{i_0}^{i'}, \tilde{e}_{l_0}^{l'}) + lb'(e_{i'+1}^{i_1}, \tilde{e}_{l'+1}^{l_1}) & \text{case (a)} \\ lb'(e_{i_0}^{i'}, \tilde{e}_{l'+1}^{l_1}) + lb'(e_{i'+1}^{i_1}, \tilde{e}_{l_0}^{l'}) + c_{INV} & \text{case (b)} \end{cases} \tag{4.31}$$

These inequalities can be used to progressively estimate the residual costs at a node in the search tree. If there estimates prove that the particular branch in the search tree starting with this node cannot be part of an optimal solution, the whole branch will be pruned.

Algorithm 4.4 is a **Memoization** approach [Michie 68, Cormen & Leiserson$^+$ 90] exploiting these estimations. In this algorithm, several upper and lower bounds are calculated, stored, and updated:

- ub_{total} stores a global upper bound for $\mathrm{d_{inv}}(e_1^I, \tilde{e}_1^L)$. This upper bound is initialized to $\mathrm{d_{LEV}}$, according to Inequality (4.26). It can be updated in runtime at the minimization step for the $\mathrm{calculateQ}(1, I; 1, L)$ node in the recursive calculation tree, at ✣ in Algorithm 4.4; for clarity, this step is not shown in the listing.
- lb_{outer} contains a lower bound to $\mathrm{d_{inv}}(e_1^I, \tilde{e}_1^L) - \mathrm{d_{inv}}(e_{i_0}^{i_1}, \tilde{e}_{l_0}^{l_1})$.
- lb_{inner} contains a lower bound to $\mathrm{d_{inv}}(e_{i_0}^{i_1}, \tilde{e}_{l_0}^{l_1})$, and finally,
- ub' stores an upper bound to $\mathrm{d_{inv}}(e_{i_0}^{i_1}, \tilde{e}_{l_0}^{l_1})$,

The main algorithm, Algorithm 4.4, initializes ub_{total} and calls Algorithm 4.5 for the whole range of i and j. This subroutine, $\mathrm{calculateQ}(e_{i_0}^{i_1}, \tilde{e}_{l_0}^{l_1})$, tests first whether $(e_{i_0}^{i_1}, \tilde{e}_{l_0}^{l_1})$ has been *memoized* before; if so, the stored value will be returned. Otherwise, it tests whether the given node belongs to deletions, insertions, substitutions, or identity operations only. If this is the case, the return value can easily be calculated and stored. If not, and if no estimation disproves the optimality of the node, the search descends into each pair of possible *straight* or *inverted* split points, by calling $\mathrm{calculateQConcatenation}$ or $\mathrm{calculateQInversion}$ (Algorithm 4.6). In these subroutines, the search algorithm uses several estimations of upper and lower bounds to prune the search tree whenever possible again. Furthermore, the algorithm *memoizes* each returned value of Q.

Although these estimation and pruning steps accelerate the calculation of $\mathrm{d_{inv}}$ significantly, they do not affect the optimality of the algorithm.

Algorithm 4.4 Memoization algorithm for the inversion distance.

$\mathbf{d_{inv,memo}}(e_1^I, \tilde{e}_1^L)$:
  $\mathrm{ub}_{total} \leftarrow \mathrm{d_{LEV}}(e_1^I, \tilde{e}_1^L)$ /* *Globally* */
  return $\mathrm{calculateQ}(1, I; 1, L)$ /* *See Algorithm 4.5* */

4.5 Word-dependent substitution costs

Traditionally, edit distances penalize word substitutions independent of whether the substituted words have a rather similar meaning (e.g., "talk"/"talks") or an absolutely different one (e.g., "talk"/"listen"). This is counter-intuitive, as replacing a word with another one with a similar meaning will rarely change the meaning of a sentence significantly, whereas replacing the same word with a completely different one probably will. Therefore, it seems advisable to make substitution costs dependent on the semantical and/or syntactical dissimilarity of the words.

The question is then how to measure this dissimilarity. A pragmatic approach is to compare the spelling of the words to be substituted with each other. The more similar the spelling is, the more similar the words are considered to be, and the lower are the substitution costs between them. This works well with similar tenses of the same verb, or with genitives or plurals of the same noun. Character-wise comparison works well with languages such as German, where verb prefixes can change, or can be split from

Algorithm 4.5 Memoization algorithm for the inversion distance: Recursion.

calculateQ$(i_0, i_1; l_0, l_1)$:
  if $i_1 < i_0$ **then**
    return $(l_1 - l_0 + 1) \cdot c_{\mathrm{INS}}$ /* *Insertion* */
  else if $l_1 < l_0$ **then**
    return $(i_1 - i_0 + 1) \cdot c_{\mathrm{DEL}}$ /* *Deletion* */
  else if $Q(i_0, i_1; l_0, l_1)$ memoized **then**
    return $Q(i_0, i_1; l_0, l_1)$
  else if $i_0 = i_1 \wedge l_0 = l_1$ **then**
    return $Q(i_0, i_1; l_0, l_1) \leftarrow (1 - \delta(e_{i_0}, \tilde{e}_{l_0})) \cdot c_{\mathrm{SUB}}$ /* *Substitution/Identity* */
  else
    $lb_{\mathrm{outer}} \leftarrow d_{\Delta \mathrm{I}}(e_1^{i_0-1} e^{\mathrm{I}}_{i_1+1}, \tilde{e}_1^{l_0-1} e^{\mathrm{L}}_{l_1+1})$
    $ub' \leftarrow ub_{\mathrm{total}} - lb_{\mathrm{outer}}$
    $lb_{\mathrm{inner}} \leftarrow d_{\Delta \mathrm{I}}(e^{i_1}_{i_0}, \tilde{e}^{l_1}_{l_0})$
    if $ub' \leq lb_{\mathrm{inner}}$ **then**
      return $Q(i_0, i_1; l_0, l_1) \leftarrow ub'$ /* *prune* */
    end if
    $lb_{\mathrm{outer}} \leftarrow d_{\mathrm{PER}}(e_1^{i_0-1} e^{\mathrm{I}}_{i_1+1}, \tilde{e}_1^{l_0-1} e^{\mathrm{L}}_{l_1+1})$
    $ub' \leftarrow ub_{\mathrm{total}} - lb_{\mathrm{outer}}$
    if $ub' \leq lb_{\mathrm{inner}}$ **then**
      return $Q(i_0, i_1; l_0, l_1) \leftarrow ub'$ /* *prune* */
    end if
    $lb_{\mathrm{inner}} \leftarrow d_{\mathrm{PER}}(e^{i_1}_{i_0}, \tilde{e}^{l_1}_{l_0})$
    if $ub' \leq lb_{\mathrm{inner}}$ **then**
      return $Q(i_0, i_1; l_0, l_1) \leftarrow ub'$ /* *prune* */
    end if
    $q_{min} \leftarrow ub'$
    for $i' \leftarrow i_0$ **to** i_1 **do**
      for $l' \leftarrow l_0$ **to** l_1 **do**
        $q_{concat} \leftarrow \mathrm{calculateQConcatenation}(i_0, i', i_1; l_0, l', l_1; ub')$
        $q_{inversion} \leftarrow \mathrm{calculateQInversion}(i_0, i', i_1; l_0, l', l_1; ub')$
        $q_{min} \leftarrow \min\{q_{min}, q_{concat}, q_{inversion}\}$ /* ✤ — *See text* */
      end for
    end for
    return $Q(i_0, i_1; l_0, l_1) \leftarrow q_{min}$
  end if

Algorithm 4.6 Memoization algorithm for the inversion distance: Inner loop.

calculateQConcatenation$(i_0, i', i_1; l_0, l', l_1; ub')$:
  $lb_1 \leftarrow d_{\Delta I}(e_{i_0}^{i'}, \tilde{e}_{l_0}^{l'})$
  $lb_2 \leftarrow d_{\Delta I}(e_{i'+1}^{i_1}, \tilde{e}_{l'+1}^{l_1})$
  if $ub' > lb_1 + lb_2$ **then**
    $lb_1 \leftarrow d_{PER}(e_{i_0}^{i'}, \tilde{e}_{l_0}^{l'})$
    if $ub' > lb_1 + lb_2$ **then**
      $lb_2 \leftarrow d_{PER}(e_{i'+1}^{i_1}, \tilde{e}_{l'+1}^{l_1})$
      if $ub' > lb_1 + lb_2$ **then**
        $lb_1 \leftarrow calculateQ(i_0, i'; l_0, l')$
        if $ub' > lb_1 + lb_2$ **then**
          $lb_2 \leftarrow calculateQ(i'+1, i_1; l'+1, l_1)$
        end if
      end if
    end if
  end if
  return $lb_1 + lb_2$

calculateQInversion$(i_0, i', i_1; l_0, l', l_1; ub')$: analogously.

the predicate. Spelling mistakes and spelling differences — for example, from American English to British English — are another point where comparing letters can be advisable. Nevertheless, it is vital to keep in mind that small spelling differences are no guarantee for a similar meaning, because prefixes such as "`mis-`", "`in-`", or "`un-`" can change the meaning of a word dramatically.

An obvious way of comparing the spelling is, again, the **Levenshtein distance**. Here, words are compared on the character level. For normalization of this distance to a range from 0 (for identical words) to 1 (for completely different words), the absolute distance is divided by the length of the Levenshtein alignment path.

Another character-based substitution cost function is based on the **common prefix length** of both words. This idea is based on the observation that in English, as well as in German or other languages, different tenses of the same verb share the same prefix; which is usually the stem. The same observation holds for different cases, numbers and genii of most nouns and adjectives in western languages. However, it does not hold when verb prefixes are changed or removed. But for the same reason, it is sensitive to critical prefixes such as "`mis-`". The length of the common prefix is normalized by the average length of both words. To achieve costs, this fraction is then subtracted from 1. Table 4.2 gives an example of these two word dependent substitution costs.

More sophisticated methods could be considered for word dependent substitution costs as well. Examples of such methods would be **stemming**, the treatment of **synonyms** and similar words using a separately trained lexicon [Vogel & Nießen$^+$ 00], or the introduction of **information weights** as in the NIST measure (see Section 3.2). However, none of these methods have been implemented for this work. Neither have been word dependent insertion or deletion costs, which could be based on information weights, for example. But independent of what is taken as cost function – for most presented algo-

Table 4.2: Example of word dependent substitution costs.

e	$\tilde{e}$	d_{LEV}	cpl*	$c_{SUB,L}$	$c_{SUB,prefix}$
`usual`	`unusual`	2	1	$\frac{2}{7} = 0.29$	$1 - \frac{1}{6} = 0.83$
`understanding`	`misunderstanding`	3	0	$\frac{3}{16} = 0.19$	1.00
`talk`	`talks`	1	4	$\frac{1}{5} = 0.20$	$1 - \frac{4}{4.5} = 0.11$
`zusagen`	`sagen`	2	0	$\frac{2}{7} = 0.29$	1.00

*common prefix length

rithms it is vital that the cost triangular inequality 4.10 must hold for all combination of words.

4.6 Overview over distance measures

Table 4.3 gives an overview of the distance measures and the algorithms to calculate them, as presented in this work. Except for $d_{\overline{CD}CD}$, all distances are symmetric if the costs are symmetric, that is, $c_{DEL} = c_{INS}$. And except for d_{inv}, all measures are also triangular. Isolation holds for all measures, although only on count vectors for d_{PER}, and only on lengths for $d_{\Delta I}$. Thus all these measures except for $d_{\overline{CD}CD}$ and d_{inv} are measures in the mathematical sense. Convexity holds for all distances. This is especially important for their use in the context of MT evaluation. The complexity of the measures differs strongly: Beginning with constant time for $d_{\Delta I}$, d_{PER} has a linear time algorithm at constant costs, whereas d_{LEV} and $d_{\overline{CD}CD}$ have quadratic time. Finally, d_{inv} is bicubic, and the full d_{LJ} calculation, in the implementation presented here, needs exponential time. The usefulness for MT evaluation in terms of correspondence with human evaluation will be the subject of the experiments in Chapter 8.

Table 4.3: Overview of the presented distance measures and algorithms.

Measure	symmetric* Equation (4.4)	triangular Equation (4.5)	convex Equation (4.6)	Time ($I \geq L$)	Space ($I \geq L$)
$d_{PER}, c = const$	✓	✓ †	✓	$O(I)$	$O(I)$
$d_{PER}, c(E, \tilde{E})$	✓	✓ †	✓	$O(I^3)$	$O(I^2)$
$d_{\Delta I}$	✓	✓ ‡	✓	$O(1)$	$O(1)$
d_{LEV}	✓	✓	✓	$O(I \cdot L)$	$O(I)$
d_{LJ}	✓	✓	✓	$O(I^2 \cdot 2^L)$	$O(I \cdot 2^L)$
$d_{\overline{CD}CD}$		✓	✓	$O(I \cdot L)$	$O(I)$
d_{inv}	✓		✓	$O(I^3 \cdot L^3)$	$O(I^2 \cdot L^2)$

*On symmetric costs
†On count vectors only
‡On lengths only

Chapter 5

Preprocessing, normalization, and reference lengths

In this chapter, preprocessing methods, normalization schemes, and similar problems common to several automatic evaluation measures are investigated. Improvements to state-of-the-art methods are given.

Several details must still be specified for the implementation and use of an automatic evaluation measure. Among others topics, the following topics require special attention: The first detail that has to be defined more precisely is the term "word" in the formulae of the previous two chapters. A common approach here for western languages is to consider spaces as **separators** of words. At these separators, the **tokenization** of a sentence into tokens (i.e., words) takes place. The role of punctuation marks in tokenization is arguable though: A punctuation mark can separate words, it can be part of a word, or it can be a word of its own. Equally, it can be irrelevant at all for evaluation. On the same lines it has to be specified whether words are considered to be different if they differ with respect to upper and lower case only. For the IWSLT evaluation, [Paul & Nakaiwa$^+$ 04] give results on how the handling of punctuation and case information may affect automatic MT evaluation. Moreover, for the automatic evaluation measures introduced here, a method to calculate the **reference length** must be specified if there are multiple reference sentences of different length.

The purpose of this work is to compare automatic evaluation with human evaluation. Therefore, two questions about human evaluation have to be clarified as well: Large evaluation tasks are usually distributed to several human evaluators. To smooth evaluation noise in an evaluation campaign, it is common practice to have at least two human judges evaluate each candidate sentence independently of each other. Therefore there are several evaluation scores for each candidate sentence. A single score for each system is required, though. Consequently, a specification is required of how the evaluator scores are combined into sentence scores, and of how these sentence scores are combined into a system score then. Different definitions here can have a significant effect on automatic and human evaluation scores.

5.1 Tokenization and punctuation marks

In written text, the importance of punctuation for the readability of a text depends on the language it is written in. So does the strictness of punctuation rules in this language. In most western languages, correct punctuation can vastly improve the legibility of texts. Marks such as full stop or comma separate sentences and words. Other marks such as apostrophes and hyphens can be used to join words, forming new words by this. For example, the spelling "There's" is a **contraction** of "There is". Similar phenomena can be found in other languages, although the set of critical characters may vary. Even when evaluating translations to English, the candidate sentences may contain source language parts such as proper names which should thus be tokenized according to rules of the source language.

From the viewpoint of the automatic evaluation measures, a decision must be taken on which units are considered to be words of their own, which are inseparable part of words, and which are irrelevant for evaluation at all. Four tokenization methods have been studied in this work: The simplest method is keeping the original sentences, and taking only spaces as word separators. Moreover, all punctuation marks can be considered to separate words, but then be removed completely in the tokenization. Version 11a of the `mteval` tool [Papineni 02] improves this scheme by keeping all punctuation marks except for decimal points and hyphens joining composita as separate words. For this study, the `mteval` scheme has been extended by the treatment of common English contractions. Figure 5.1 illustrates these methods.

5.2 Case sensitivity

In western languages, maintaining correct upper and lower case can improve the readability of a text. Unfortunately, though the case of a word depends on the word class, classification is not always unambiguous. What is more, the first word in a sentence is always written in upper case. This lowers the significance of case information in MT eval-

- Original candidate
  `Powell said: "We'd not be alone; that's for sure."`
- Remove punctuation
  `Powell said We d not be alone that s for sure`
- Tokenization of punctuation (`mteval`)
  `Powell said : " We'd not be alone ; that's for sure . "`
- Tokenization and treatment of abbreviations and contractions
  `Powell said : " we would not be alone ; that is for sure . "`

Figure 5.1: Tokenization methods studied in this work. Each underlined character sequence corresponds to a "word" in the sense of the evaluation measures.

uation, as even a valid reordering of words between candidate and reference sentence may lead to conflicting cases. Consequently, the present study included experiments on whether and how case information can be exploited for automatic evaluation.

5.3 Reference length calculation

Each automatic evaluation measure taken into account for this work depends on the calculation of a **reference length**: The reference length normalizes WER, PER, and ROUGE, whereas NIST or BLEU incorporate it for the determination of the brevity penalty. In MT evaluation practise, there are multiple reference sentences for each candidate sentence, with different lengths each. It is thus not intuitively clear what this *"reference length"* is.

5.3.1 Average length

A simple choice for the reference length is the average length of the reference sentences. With R_k as the number of reference sentences for candidate sentence E_k, and $L_{r,k}$ as the length of reference sentence $\widetilde{E}_{r,k}$, this is:

$$L_k^* := \frac{1}{R_k} \sum_r L_{r,k} \tag{5.1}$$

Though this is modus operandi for the NIST measure [Doddington 02], the average length can be problematic with a score based on the F-measure or a brevity penalty, because even candidate sentences that are identical to a shorter-than-average reference sentence, which would intuitively be considered as "optimal", will then receive sub-optimal scores.

5.3.2 Minimum nearest length

In its default implementation by [Papineni 02], BLEU incorporates a different method for the determination of the reference length: Reference length here is the reference sentence length which is closest to the candidate length. If there is more than one, the shortest of these lengths is chosen:

$$L_k^* := \min\left\{ L_{r,k} \;\middle|\; |I_k - L_{r,k}| = \min_{r'} |I_k - L_{r',k}| \right\} \tag{5.2}$$

5.3.3 Average length of nearest sentences

For measures based on the comparison of single sentences, such as WER, PER, and ROUGE, another method deserves consideration: The average length of the sentences with the lowest absolute distance or highest similarity to the candidate sentence.

$$L_k^* := \frac{1}{|\mathcal{R}_k'|} \sum_{r \in \mathcal{R}_k'} L_{r,k} \tag{5.3}$$

with

$$\mathcal{R}'_k := \left\{ r \mid d\big(E_k, \widetilde{E}_{r,k}\big) = \min_{r'} d\big(E_k, \widetilde{E}_{r',k}\big) \right\} \tag{5.4}$$

This method is called "average nearest sentence length".

5.3.4 Length of best sentence

This reference length determination method takes the length of the sentence with the lowest relative error rate or the highest relative similarity. With the "best" reference sentence

$$r^*_k := \operatorname*{argmin}_r \frac{d\big(E_k, \widetilde{E}_{r,k}\big)}{L_{k,r}} \tag{5.5}$$

the reference length is

$$L^*_k := L_{r^*_k,k} \tag{5.6}$$

When using this method, Equation (4.1) must be altered as follows:

$$ER := \frac{1}{L^*_{tot}} \sum_k d\big(E_k, \widetilde{E}_{r^*_k,k}\big) \tag{5.7}$$

5.3.5 Other methods for the calculation of a reference length

Other strategies for the determination of a reference length have been studied; for example, the maximum or minimum reference sentence length. None of these methods showed a theoretical or experimental advantage over the first four methods. Nor could any reference to an actual usage of any of these methods in MT evaluation practice be found.

5.4 Sentence Boundaries

The position of a word within a sentence can be rather significant for the correctness of the sentence. WER, INVWER, and ROUGE-L take the ordering into account explicitly. This is not the case with m-PER, BLEU, or NIST, although the positions of inner words have an implicit relevance by the m-gram overlap. To model the position of words at the initial or the end of a sentence with the same importance, artificial sentence boundary words "<s>", "</s>" can enclose the sentence. Although this is a common approach in language modeling, MT researchers have to the author's knowledge not yet applied it to MT evaluation. Figure 5.2 gives an example of such boundary-enhanced bigrams and trigrams.

Notice that the application of this scheme to the NIST measure creates an additional problem regarding the sentence initial m-gram: According to Equation (3.8), the count of the sentence initial $(m-1)$-gram [<s> ... <s>] is needed for the calculation of the information weight for the sentence initial m-gram [<s> ... <s> e_1]. This count is

Sentence: `I prefer the plane`
Bigrams: `[<s> I]`, `[I prefer]`, `[prefer the]`, `[the plane]`, `[plane </s>]`
Trigrams: `[<s> <s> I]`, `[<s> I prefer]`, `[I prefer the]`, `[prefer the plane]`, `[the plane </s>]`, `[plane </s> </s>]`

Figure 5.2: Example of artificial sentence boundaries.

zero by definition, which would cause the information weight to be undefined. To avoid an undefined information weight, the definition of the counts has been modified such that each initial $(m-1)$-gram is assumed to appear once in each sentence; that is,

$$\mathit{Info}\big(\texttt{[<s> ... <s> } e_1 \texttt{]}\big) := -\big(\log_2 \widetilde{N}_{\texttt{[<s> ... <s>]}} - \log_2 K\big) \tag{5.8}$$

For all other parts of the NIST formula, the initial $(m-1)$-gram `[<s> ... <s> ]` is considered to be inexistent.

5.5 Evaluator normalization for human evaluation

For human evaluation, it has to be specified how **evaluator bias** is coped with, and how sentence scores are combined into system scores. Regarding evaluator bias, even accurate evaluation guidelines will not prevent a measurable discrepancy between the scores assigned by different human evaluators. The 2003 TIDES/MT evaluation [Przybocki 03] (see also Section 7.1.2) may serve as an example here: One would expect the assessed scores to be independent of the evaluator, because the candidate sentences of the participating systems were randomly distributed among ten human evaluators. Figure 5.3 indicates that this is indeed not the case, as the evaluators can clearly be distinguished on the amount of good and bad marks they have assessed.

[Doddington 03] proposed $(0,1)$ **evaluator normalization**, which overcomes this bias: For each human evaluator the average sentence score given by him or her and the variance of this score are calculated. These assignments are then normalized to $(0,1)$ expectation and standard deviation, separately for each human evaluator.

Although this normalization is important for evaluation on sentence level, system level evaluation does not require such a step: If the distribution of evaluators and systems is random enough, the evaluator biases tend to cancel out over the large amount of candidate sentences. Moreover, with $(0,1)$ normalization the calculated system scores are relative scores, rather than absolute scores. As such they can be compared with scores from the same evaluation only.

Usually, there are several assessments from different evaluators for each candidate sentence. Only one score is required per sentence, though. Depending on the number of these assessments, different combination methods for the sentence scores can be considered; their mean or their median, for example. As the test data for this work consisted of only two or three human assessments per sentence, only the arithmetic mean was applied in the experiments conducted for this study.

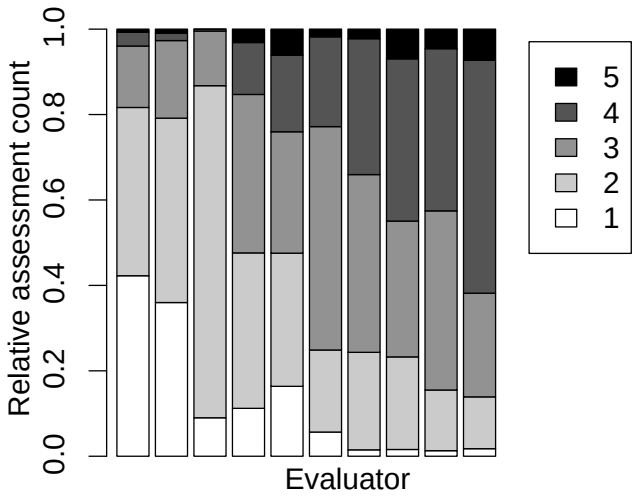


Figure 5.3: Distribution of adequacy assessments (1–5) for each human evaluator. TIDES 2003 CE corpus.

Whereas the assessments of the human evaluators are given on the sentence level, the interest of the evaluation campaign organizer will often lie on the evaluation of whole candidate systems. Therefore it must be defined how a system score is calculated from the sentence scores. All automatic evaluation measures implicitly weight the candidate sentences by their length. Consequently, in the system level experiments for this work, the sentence scores were weighted by the sentence length as well.

Chapter 6

Correlation

In this chapter, different coefficients for the quantitative evaluation of the correlation between two probability distributions are described. Automatic MT evaluation scores on a test set can be regarded as samples of a probability experiment as well. Consequently, correlation coefficients can give evidence of the correspondence between automatic and human MT evaluation measures as well.

For the reasons listed in Section 1.2.2, an automatic evaluation measure for MT is considered useful if it corresponds well with human evaluation. To compare the quality of different evaluation measures, preprocessing methods, or evaluation parameters, this correspondence needs to be measured quantitatively. The approach taken in this work is to measure correlation empirically: On a test set of candidate translations generated by different MT systems, automatic evaluation scores are compared with human evaluation scores for these candidate translations. That these candidate translations were generated by MT systems increases the likeness to the environment where the automatic evaluation measures will be used in practice. The technique of the evaluation of automatic MT evaluation measures by calculating the correlation with human evaluation has been applied before by [Lin & Och 04a, Doddington 02, Papineni & Roukos$^+$ 02] and others.

For a statistical analysis of MT evaluation measures, all these measures – human and automatic – are treated as aligned random variables. The individual scores can then be regarded as aligned random samples for these variables. These samples can be sentence scores, document scores, or system scores, depending on whether the granular correlation, or the overall stability of the measure is to be assessed. For pairs of random variables, several correlation coefficients have been defined: Among others, the most important are **Pearson's** r, **Spearman's** ρ, and **Kendall's** τ. This chapter will cover these three correlation coefficients.

Direct comparison with human evaluation is not the only possible way to evaluate the expressiveness of automatic evaluation measures: [Lin & Och 04b] define the ORANGE scheme, which does not require any explicit human evaluation. In addition to the set of multiple reference sentences required by the automatic evaluation measures, only a preferably large set of non–perfect candidate translations is required. In their experiments, Lin and Och used n-best-lists of their MT system for this. The principal idea behind the ORANGE scheme is that they expect these machine translations to be worse

than the provided reference translations in terms of quality. A requirement for an automatic evaluation measure is that it should be able to decide between "perfect" reference translations and "non–perfect" candidate translations. The **ranks** that the measure assigns to reference translations within the set of candidate translations give a measure to that ability. The lower the ranks of the known-perfect reference translation are according to the automatic evaluation measure, the worse the quality of this very measure is considered to be. Reference sentences must not compared with themselves for the ranking, as they would otherwise receive best scores. Therefore, cross validation or leaving-one-out [Efron & Tibshirani 93] must take place.

The ORANGE meta evaluation scheme stands and falls with the quality and structure of both reference and candidate translations – good candidate translations and bad reference translations will result in a bad ORANGE score for each measure, whereas bad MT and good reference sentences are likely to overrate the measures. Although the ORANGE scheme seems interesting enough to require further examination, this work will cover only correlation experiments.

6.1 Pearson's r

6.1.1 Definition

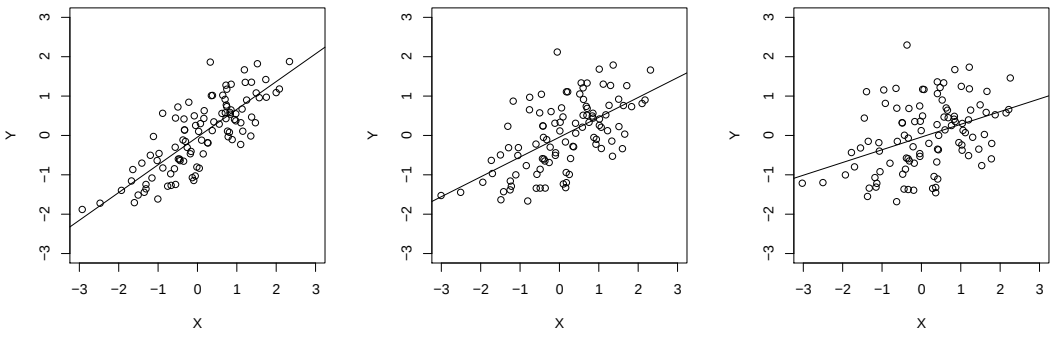
Pearson's correlation coefficient r [Casella & Berger 90] is based on the idea that two linearly correlated random variables show a high covariance with each other. To render the coefficient independent of the actual variance of the variables themselves, the covariance is normalized by the standard deviation of both variables:

$$r_{XY} := \frac{\sigma_{XY}}{\sigma_X \sigma_Y} \tag{6.1}$$

With this normalization, r_{XY} becomes 1 exactly if X and Y show perfect linear positive correlation (i.e., all (x, y) pairs lie exactly on an ascending line), and it becomes -1 if X and Y show perfect linear negative correlation. If there is no linear correlation between the variables at all, r_{XY} becomes zero. An example of distributions with different values of r can be found in Figure 6.1. Except for the sign, Pearson's r is invariant to scale and translation; that is,

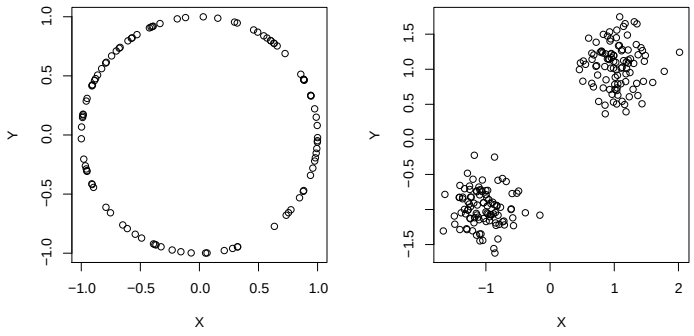
$$r_{XY} = r_{XY'} \qquad \text{where } Y' = aY + b \text{ with } a > 0 \tag{6.2}$$

Note that r covers only linear correlation; a quadratic or an even more difficult correlation may or may not be detected well. For example, $r_{XY} = 0$ in Figure 6.2(a). Furthermore, large scale correlation dominates local correlation. The two distributions in Figure 6.2(b) have a rather high correlation coefficient of $r_{XY} = 0.92$, despite the obvious variance between them.



(a) $r_{XY} = 0.8$ (b) $r_{XY} = 0.6$ (c) $r_{XY} = 0.4$

Figure 6.1: Example of probability distributions for different values of Pearson's correlation coefficient r.



(a) "Circular" correlation. $r_{XY} = 0$ (b) Large scale correlation. $r_{XY} = 0.92$

Figure 6.2: Example of nonlinearly and large-scale correlated variables.

6.1.2 r and linear regression

In addition to its theoretical definition, Pearson's correlation coefficient has a graphical meaning as well. For linear regression, r serves as an indicator for the quality of the regression; more precisely for the regression error:

Let X and Y be random variables; let $\hat{Y}$ be the linear regression of Y given X; that is, $\hat{y}_i = \alpha + \beta x_i$. In this equation, α and β are chosen such that the expected regression error square $E\left[\|Y - \hat{Y}\|^2\right]$ is minimal. This linear regression error is measured parallel to the Y axis, not necessarily orthogonal to the regression line. Then, this expected regression error can be calculated from r_{XY} and the variance σ_{YY} alone as follows:

$$E\left[\|Y - \hat{Y}\|^2\right] = \sigma_{YY}(1 - r_{XY}^2) \tag{6.3}$$

For a proof, see Appendix C.1. A similar proposition holds for a regression of X given Y:

$$E\left[\|X - \hat{X}\|^2\right] = \sigma_{XX}(1 - r_{XY}^2) \tag{6.4}$$

Figures 6.3(a) and 6.3(b) give examples of linear regressions and regression errors between random distributions X and Y with $r_{XY} = 0.75$

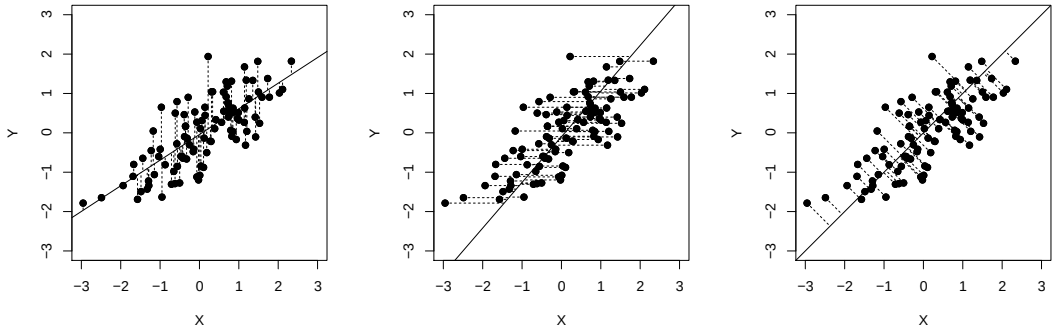
6.1.3 r and least orthogonal squares regression

Applying linear regression requires errors to occur only on one of the two variables. In practice, this is rarely the case. In the context of evaluating automatic evaluation measures for MT, human evaluators will make mistakes when evaluating sentences. Similarly, automatic evaluation measures will misjudge single sentences as well.

Instead of linear regression, a more suitable regression method will be used in such cases; namely a **Least orthogonal squares regression**. Here, the regression line $(\hat{X}, \hat{Y})$ will be chosen such that the expectation of the distance between (x_i, y_i) and the line, this time measured orthogonal to the line, is minimal. Fortunately, r_{XY} gives an indicator of the quality of this regression as well: Assume that X and Y are normally distributed, and have the same variance $\sigma_{XX} = \sigma_{YY}$. The latter is a reasonable assumption, because r is invariant to positive scalar multiplication. Moreover, most error measures are normalized to values between 0 and 1. Let $(\hat{X}, \hat{Y})$ be a least orthogonal squares regression of (X, Y). Then, for the expectation of the orthogonal squares, $E\left[\|(X, Y) - (\hat{X}, \hat{Y})\|^2\right]$, the following equation holds:

$$E\left[\|(X, Y) - (\hat{X}, \hat{Y})\|^2\right] = \sigma_{XX}(1 - |r|) \tag{6.5}$$

A proof is given in Appendix C.2. Figure 6.3(c) gives an example of a least orthogonal squares regression.



(a) Y given X (b) X given Y (c) Least orthogonal squares

Figure 6.3: Example of different linear regressions.

6.1.4 Optimal linear combination of evaluation measures regarding r

Because of the varying performance of different automatic evaluation measures, a possible approach to get "the best of all worlds" is to create a new measure as a linear combination of these evaluation measures. A higher correlation with human evaluation is the usual intention here. A mathematical analysis of this method reveals that Pearson's r between the combined automatic evaluation measures and human evaluation depends only on the covariance among the measures themselves, rather than on the actual samples. The calculation of these covariance itself requires the actual data once, but after that, no further look on the samples is necessary.

Special case: Linear combination of two evaluation measures

With two evaluation measures X_1, X_2, here written as random variables, and a linear weight $0 \leq w \leq 1$, the combined measure X_w is

$$X_w := wX_1 + (1-w)X_2 \tag{6.6}$$

Then, for the variance of X_w holds:

$$\sigma_{X_w X_w} = w^2\sigma_{X_1X_1} + 2w(1-w)\sigma_{X_1X_2} + (1-w)^2\sigma_{X_2X_2} \tag{6.7}$$

and for the covariance between X_w and evaluation measure Y:

$$\sigma_{X_w Y} = w\sigma_{X_1Y} + (1-w)\sigma_{X_2Y} \tag{6.8}$$

A derivation of this can be found in Appendix C.3. Hence, Pearson's r between X_w and Y can be calculated as:

$$r_{X_w Y} = \Big(\frac{1}{\sigma_{YY}} \frac{w^2\sigma^2_{X_1Y} + 2w(1-w)\sigma_{X_1Y}\sigma_{X_2Y} + (1-w)^2\sigma_{X_2Y}}{w^2\sigma_{X_1X_1} + 2w(1-w)\sigma_{X_1X_2} + (1-w)^2\sigma_{X_2X_2}}\Big)^{\frac{1}{2}} \tag{6.9}$$

General case: Linear combination of n evaluation measures

In the general case, there is a vector of different automatic evaluation measures $\mathbf{X} = \{X_1, \ldots, X_n\}$. Let Σ be the covariance matrix among the evaluation measures:

$$\Sigma := \begin{pmatrix} \sigma_{X_1X_1} & \cdots & \sigma_{X_1X_n} \\ \vdots & \ddots & \vdots \\ \sigma_{X_nX_1} & \cdots & \sigma_{X_nX_n} \end{pmatrix} \tag{6.10}$$

and let $\boldsymbol{\sigma}_Y$ be the covariance vector between each automatic measure and the human evaluation measure:

$$\boldsymbol{\sigma}_Y = \left(\sigma_{X_1Y}, \ldots, \sigma_{X_nY}\right)^T \tag{6.11}$$

The weight vector $\mathbf{w} = (w_1, \ldots, w_n)^T$ determines the linear combination of the measures:

$$X_{\mathbf{w}} = \mathbf{w}^T\mathbf{X} \tag{6.12}$$

Then, variance, correlation, and correlation coefficient for the combined automatic error measure are:

$$\sigma_{X_wX_w} = \mathbf{w}^T\Sigma\mathbf{w} \tag{6.13}$$

$$\sigma_{X_wY} = \mathbf{w}^T\boldsymbol{\sigma}_Y \tag{6.14}$$

and

$$r_{X_wY} = \frac{\mathbf{w}^T\boldsymbol{\sigma}_Y}{\sqrt{\mathbf{w}^T\Sigma\mathbf{w} \cdot \sigma_{YY}}} \tag{6.15}$$

It may be surprising that the correlation can already be calculated from the covariance matrix Σ between the automatic measures, the covariance vector σ_Y between the automatic measures and the human measure, and the variance of the human measure. Even a correlation-optimal weight vector $\mathbf{w}^*$ can be found using only these covariances. It is still important to be careful when adjusting parameters – as this weight vector – on test data; especially, as these parameters depend largely on the data itself. At least, cross validation checks or similar steps must be taken to avoid training on test data.

6.2 Spearman's ρ

In spite of the wide-spread use of Pearson's r in the area of evaluating automatic evaluation measures for MT, there are several problems connected with it. First of all, the researcher is not interested in the absolute value of the measure in most applications, but in relative rankings of different candidate sets; for example, she might be interested which of several MT systems produces the best output, or what value for a parameter is optimal in terms of the generated translations. Anyhow, r weights large scale correlation much higher than small scale variations. Connected with this is the problem that r is

designed to provide information about linear correlation only. Nonlinear correlation is addressed insufficiently in many cases.

Moreover, there is the mathematical problem that Pearson's correlation coefficient is a **parametric** method; that is, it requires the random variables to be normally distributed. This is, strictly speaking, not the case for automatic and human evaluation measures, especially when evaluating on sentence level – five different outcomes for the human evaluation of adequacy can hardly be declared to form a normal distribution.

For this, several **nonparametric** correlation coefficients have been defined [Kendall 70, Siegel & Castellan 88]. As these coefficients take only the rank of a sample among the other samples into account, instead of its actual value, they are called **rank correlation coefficients**.

A simple rank correlation coefficient is **Spearman's** ρ, which is basically *Pearson's* r on ranks: Let X and Y be aligned random variables with samples $(x_1, y_1), \ldots, (x_n, y_n)$. Let $\mathtt{rank}_X, \mathtt{rank}_Y$ be the ranks of the x_i, y_i, that is,

$$\left(\mathtt{rank}_{X,i} < \mathtt{rank}_{X,j}\right) \;\Leftrightarrow\; \left(x_i < x_j\right) \tag{6.16}$$

and

$$\forall i : \mathtt{rank}_{X,i} \in \{1, \ldots, n\} \tag{6.17}$$

The same holds for $\mathtt{rank}_Y$. Ties are treated using mid-ranking. Then,

$$\rho_{XY} := r_{\mathtt{rank}_X \mathtt{rank}_Y} \tag{6.18}$$

If there are no ties, this is

$$= 1 - 6 \cdot \frac{\sum_i (\mathtt{rank}_{X,i} - \mathtt{rank}_{Y,i})^2}{n} \tag{6.19}$$

As a rank correlation coefficient, ρ_{XY} becomes 1 exactly if X and Y show perfect monotonous positive correlation, and -1 if they show perfect monotonous negative correlation. A ρ_{XY} of 0 indicates that no monotonous correlation can be found. Linearity is no longer a requirement to the correlation measured using the coefficient – strict monotonicity suffices.

6.3 Kendall's τ

6.3.1 Definition

Although Spearman's rank correlation coefficient has a clear and concise definition, it is rather indescriptive. Therefore, this study uses another rank correlation coefficient, namely **Kendall's rank correlation coefficient** τ. Basically, for the two variables (or measures), τ denotes the empirical probability of agreement on random samples minus the probability of disagreement: With $1 \leq i, j \leq N$, let

$$C := \left| \{ i < j \mid x_i < x_j \text{ and } y_i < y_j \qquad \text{or vice versa} \} \right| \tag{6.20}$$

$$D := \left| \{ i < j \mid x_i < x_j \text{ and } y_i > y_j \qquad \text{or vice versa} \} \right| \tag{6.21}$$

$$M := |\{ i < j \}| \; = \; \frac{N/ \cdot (N-1)}{2} \tag{6.22}$$

Then,

$$\tau_{XY} := \frac{C - D}{M} \tag{6.23}$$

Again, a τ_{XY} of 1 denotes that X and Y show perfect monotonous positive correlation, and a τ_{XY} of -1 indicates a perfect monotonous negative correlation.

6.3.2 τ on sentence level: $\overline{\tau}$

On a low or moderate number of samples, as the comparison of human and automatic evaluation on system level is, Kendall's τ is a reasonable measure for the correlation. Comparison of human and automatic evaluation on sentence level leads to two problems that require a modification: First of all, the number of sentence is in the order of several thousands to tens of thousands. Several statistical methods require vast amounts of correlation coefficients; **bootstrapping** [Efron & Tibshirani 93, Bisani & Ney 04] being an example. As the computational complexity of Kendall's τ is higher than the same of Pearson's r and Spearman's ρ, this may form an obstruction to the use τ in these methods. Anyhow, the more serious problem when calculating Kendall's τ on automatic and human MT evaluation scores is the large number of ties. The low number of different possible outcomes in human sentence evaluation – typically five – aggravates this problem, because this means that at least 20% of all pairs (i, j) in Equation (6.23) are ties.

In most applications, the ability of an evaluation measure to rank candidate translations of *different* source sentences produced by the *same* MT system is of less importance than the ability to rank candidate translations of the *same* source sentence produced by *different* MT systems $s = 1, \ldots, S$. Consequently, not the rank correlation over all candidate translations and all MT systems is asked for, but the **local rank correlation** over the different MT systems for each source sentence separately. This can then be averaged over the whole set of candidate sentences:

$$\overline{\tau} := \frac{1}{S} \sum_{s} \tau_{X_s Y_s} \tag{6.24}$$

Chapter 7

Corpora for the evaluation of automatic MT evaluation

This chapter gives an overview over the corpora used for the experiments in this study.

For the experimental assessment of the different automatic evaluation measures in this study, a large set of candidate translations was required. For each candidate translation, human evaluation scores as well as reference translations were necessary. Fortunately, several international evaluation campaigns have been held during the last three years. For these competitions, the state-of-the-art MT systems of the participating research groups had to translate a set of 500 to 1500 Chinese, Japanese, or Arabic sentences. A group of independent human judges then evaluated the candidate translations. Each sentence was presented to human judges, who assigned a score from 1 (worst) to 5 (best) for both fluency and adequacy, as listed in Table 7.1. To minimize a possible evaluator bias, at least two independently chosen evaluators assessed each sentence. Additionally, a ranking using established automatic evaluation measures such as WER or BLEU took place. Most participants of the campaigns agreed to the publishing of their (anonymized) candidate translations and the human assessments for them. Consequently, data from these campaigns constitutes useful corpora for the comparison of automatic evaluation measures with human evaluation.

Seven corpora were used for the experiments in this study. The corpus statistics of these corpora are listed in Table 7.2. Only data relevant for the experiments are accounted for; candidate sentences that have no human evaluation are not included in the figures. Experiments for the measures with a high complexity, namely INVWER and LJWER, forbade sentence lengths exceeding 20. Therefore, reduced test sets were created consisting only of source sentences where each reference and candidate translation had a length of 20 words or shorter. The size of these reduced corpora are also listed in Table 7.2. Experiments that neither involved INVWER nor LJWER were conducted on the full test corpora.

Table 7.1: Description of different fluency and adequacy scores.From [LDC 05].

Score	As **adequacy** judgment: Information of original sentence present in candidate sentence:	As **fluency** judgment: Linguistic quality of candidate sentence:
5	All	Flawless
4	Most	Good
3	Much	Non–native
2	Little	Disfluent
1	None	Incomprehensible

Table 7.2: Corpus statistics.

	TIDES 2002	TIDES 2003	TIDES 2003	TIDES 2004	TIDES 2004	BTEC 2004	BTEC 2004
Source language	Chinese	Chinese	Arabic	Chinese	Arabic	Chinese	Japanese
Target language	English	English	English	English	English	English	English
Sentences	878	919	663	446	347	500	500
Sent. $\leq$ 20 words	169	273	142	64	75	477	483
Running words	24 084	25 784	17 763	13 016	10 892	3 632	3 632
Punctuation marks	2829	3215	2343	1516	1242	610	610
Ref. translations	4	4	4	4	4	16	16
Avg. ref. length	27.4	28.1	26.8	29.2	31.4	7.3	7.3
Candidate systems	9	7	6	10	5	11	8
Case information*		i?	i?	u?	u?	i	i

*for human evaluation; *i*: ignore case, *u*: use case.

7.1 NIST/TIDES

7.1.1 TIDES 2002 Chinese–English

From the *Translingual Information Detection, Extraction, and Summarization Project* (TIDES), 70 newswire stories from Xinhua News Service and 30 web news stories from Zaobao News Agency over the period of 1994 to 1998 were selected as a test set for the Chinese–English task of the TIDES 2002 MT evaluation workshop [NIST 02, Ciery & Huang$^+$ 02]. Table 7.3 lists the share of both sources on this corpus. To achieve a set of reliable reference sentences for the automatic evaluation measures, several different professional translation agencies translated these texts. A human expert selected the best four translations each as reference translations.

For human evaluation [LDC 05], each candidate sentence was presented to two or three out of ten English-speaking judges. These judges assessed adequacy with respect to a selected reference translation. Fluency was to be judged from the evaluators' linguistic competence only.

7.1.2 TIDES 2003 Chinese–English, Arabic–English

The NIST/TIDES 2003 evaluation workshop [Przybocki 03] consisted of an Arabic–English task and a Chinese–English task. For both tasks, 50 news stories from Xinhua News Service and 50 more news stories from Agency France Press (AFP) over the period of January and February 2003 were selected as test set. To achieve a set of reliable reference sentences, four different professional translation agencies translated these texts. Table 7.4 lists the share of both sources on both corpora.

For human evaluation [LDC 05], each candidate sentence was given to two out of ten English-speaking judges. Except for the now constant number of judges per sentence, human evaluation conditions were the same as in the 2002 task.

7.1.3 TIDES 2004 Chinese–English, Arabic–English

Both the Arabic–English and the Chinese–English task of the NIST 2004 evaluation workshop [Przybocki 04] consisted of 50 editorials, 50 speech parts, and 100 news articles from various sources over the period of November 2003 to March 2004. Speeches were collected out of a period from 2002 to 2004. An overview of the sources is given in Table 7.5. Again, different professional translation agencies provided four reference translations

Table 7.3: Sources of TIDES/NIST 2002 Chinese–English task.

	Documents	Sentences
XINHUA	70	546
Zaobao	30	332
Total	100	878

Table 7.4: Sources of TIDES/NIST 2003 Chinese–English and Arabic–English task.

	Chinese–English		Arabic–English	
	Documents	Sentences	Documents	Sentences
AFP	50	495	50	338
XINHUA	50	424	50	325
Total	100	919	100	663

for each sentence. In contrast to previous evaluation campaigns, only 446 of the 1788 Chinese–English and 347 of the 1353 Arabic–English sentences were then assessed each by two out of sixteen human evaluators. For the experiments in this study, these sentences alone were of interest; all corpus statistics in Table 7.2 refer only to these sentences.

7.2 BTEC/IWSLT

The C-STAR Consortium created a multilingual corpus consisting of short phrases and sentences from the tourism domain [BTEC 04], parallel in eight languages. This corpus was called **Basic Travelling Expressions Corpus** (BTEC). In contrast to the NIST/ TIDES corpora, sentences from the BTEC corpus are mostly written speech and personal communication.

For the 2004 IWSLT evaluation campaign [Akiba & Federico$^+$ 04], 500 Chinese and 500 Japanese sentence from the BTEC were selected as test corpus. Target language for both tasks was English. Native English speakers created up to 15 different paraphrases of the English translation provided in the original BTEC corpus. This makes a total of up to 16 reference translations for each sentence. Ten monolingual judges evaluated the candidate translations, first with regard to fluency, then with regard to adequacy by comparison with a reference sentence. In total, each candidate sentence was assessed by three judges.

Table 7.5: Sources of TIDES/NIST 2004 Chinese–English and Arabic–English task.

	Chinese–English		Arabic–English	
	Documents	Sentences	Documents	Sentences
Editorial	50	449	50	368
News	100	901	100	707
Speech	50	438	50	278
Total	200	1788	200	1353
Evaluated		446		347

Chapter 8

Experimental results

In this chapter, experimental results on the correlation between human and automatic evaluation will be presented. These experiments have been performed for different evaluation measures, preprocessing and normalization steps, and so on.

8.1 Normalization and summation of human evaluation

In each of the corpora in this study, the candidate sentences were evaluated by at least two different human evaluators each. The **Inter-Annotator Correlation** (IAC) between the different human sentence scores for each sentence is a measure for the **Inter-Annotator Agreement** — that is, for the accordance of judgments from the human evaluators. Table 8.1 shows the IAC for the different corpora. For technical reasons, only two judgments of each sentence were regarded for the IAC calculation for the TIDES 2002 task here. It can be seen that the earlier evaluation campaigns have a rather poor IAC. In later campaigns, the better-rehearsed evaluation process as well as improved evaluation guidelines seem to have improved the agreement between the different human judges. Also shown in the table is the effect of $(0,1)$-evaluator normalization, as described in Section 5.5. It can

Table 8.1: Effect of $(0,1)$-evaluator normalization on the inter-annotator correlation. Pearson's r on sentence level.

		TIDES 2002 CE*	TIDES 2003 CE	TIDES 2003 AE	TIDES 2004 CE	TIDES 2004 AE	BTEC 2004 CE	BTEC 2004 JE
A	no normalization	0.11	0.18	0.23	0.40	0.41	0.80	0.84
	$(0,1)$-normalization	0.28	0.36	0.38	0.51	0.51	-	-
F	no normalization	0.01	0.06	0.11	0.32	0.32	0.77	0.81
	$(0,1)$-normalization	0.19	0.21	0.30	0.42	0.38	-	-

*between two assessments only

easily be seen that this normalization step increases the IAC significantly. This normalization was not possible for the BTEC corpora, as the provided information did not allow a mapping between score and individual evaluators.

A lower IAC indicates a lower confidence in the human evaluation score. Consequently, for a lower IAC, a lower correlation with other evaluation measures can be expected. Table 8.2 confirms this statement: The higher the IAC is, the higher is the correlation between each human score and WER (or any other automatic evaluation measure not shown in the table). Moreover, $(0,1)$-evaluator normalization improves the correlation between human and automatic evaluation on sentence level significantly.

Within this table, In Tables 8.1 and 8.2, as well as all following result tables, the given correlation coefficient is negated for correlation between a quality measure (such as BLEU, or adequacy) and an error measure (such as WER). Therefore, a positive coefficient corresponds to the intuitive understanding of the measure, whereas a negative coefficient in the tables denotes counter-intuitive behavior of a measure. In the tables, "A" denotes the correlation with adequacy, whereas "F" denotes the correlation with fluency. The sum of adequacy and fluency serves as an overall quality measure for candidate sentences; the correlation with this score is denoted by "$A + F$". A bold numbers indicates the highest correlation within an experiment for the corpus and the human evaluation measure. The 95-percent **confidence range** for one of the values can be found in the table as well. Differences in the coefficients larger than this range can be expected to differ significantly with a confidence of 95 percent or higher. For the sake of clarity, only the most interesting parts of result tables are put down in this chapter. Complete tables can be found in Appendix B.

8.2 Baseline settings and default settings

As experimental baseline, the established MT evaluation measures WER, PER, BLEU, and the NIST measure were used. Preliminary experiments [Leusch & Ueffing$^+$ 05] have indicated that certain changes to the "baseline" settings can have a positive effect on the correlation between automatic and human evaluation. These changes include the treatment of abbreviation in addition to the "baseline" `mteval` tokenization step, the use of the length of the score-best reference sentence as reference length, and the use of sentence boundaries and $(0,1)$-evaluator normalization. Table 8.3 gives an overview on the default settings for the experiments in opposite to the "baseline" settings Detailed experiments on the particular settings can be found in the rest of this chapter.

Consequently, all experiments for this work have been conducted with these modifications to the evaluation measures and preprocessing steps, if not stated there otherwise.

The listed changes do indeed increase the correlation between human evaluation and automatic evaluation. For the correlation on sentence level, as expressed by Pearson's r, this can be seen in Table 8.4.

For a comparison of how the linear correlation between automatic and human sentence evaluation corresponds to the local ranking ability, Table 8.5 lists the average *local rank correlation* $\bar{\tau}$ between automatic and human evaluation for both baseline and default settings. Both in terms of linear correlation and local ranking, the BLEU measure and the

Table 8.2: Effect of $(0,1)$-evaluator normalization on the correlation between WER and human evaluation.Pearson's r on sentence level.

Human score	Evaluator normalization	TIDES 2002 CE	TIDES 2003 CE	TIDES 2003 AE	TIDES 2004 CE	TIDES 2004 AE
A	no normalization	0.285	0.307	0.454	0.507	0.566
	95%-Conf.	*±0.020*	*±0.022*	*±0.024*	*±0.021*	*±0.031*
	$(0,1)$-normalization	**0.320**	**0.349**	**0.505**	**0.540**	**0.597**
F	no normalization	0.223	0.243	0.374	0.480	0.457
	95%-Conf.	*±0.021*	*±0.023*	*±0.026*	*±0.022*	*±0.036*
	$(0,1)$-normalization	**0.277**	**0.301**	**0.423**	**0.511**	**0.496**
A+F	no normalization	0.282	0.306	0.453	0.529	0.554
	95%-Conf.	*±0.020*	*±0.022*	*±0.024*	*±0.021*	*±0.032*
	$(0,1)$-normalization	**0.328**	**0.365**	**0.518**	**0.559**	**0.589**

similar NIST measure correlate highest with human evaluation on the TIDES corpora. In contrast, PER judges adequacy best and WER corresponds best with fluency for the BTEC corpora. In all cases, the proposed changes in preprocessing and so on increase sentence level evaluation significantly. PER does not regard the ordering of the words in a sentence, just their occurrence. WER, on the other hand, does not allow any reordering at all. Consequently, PER correlates to a higher degree with adequacy than WER does (which is mainly an evaluation of how much of the transported information is correct), whereas WER correlates to a higher degree than PER on fluency, where the correct order of words is important.

But does this supremacy of BLEU towards WER and PER, and the prevalence of the improved methods reflect on system level MT evaluation? Table 8.6 shows that this is only partly the case: Whereas a good correlation on sentence level for most measures comes along with a good correlation on system level, and a bad correlation on the former with a bad correlation on the latter, the small amount of sample points on system level (5 to 11) takes its toll: Not only do the correlations on most corpora hardly differ significantly. Certain preprocessing methods have unequally distributed effect on single MT systems, shifting the scores of these systems more to the lower (or to the higher) side than the scores of other systems. This finding might explain why the default settings perform significantly better in terms of correlation on all corpora and all evaluation measures on sentence level than the baseline settings, but worse for all measures on the BTEC 2004 Chinese–English corpus on system level.

Furthermore, it can be seen that rankings by different evaluation measures are noticeably similar, and usually are the same for each corpus. For example, all measures except for PER rank the participating systems of the TIDES 2004 Arabic–English evaluation campaign the same as the human judges do. Nevertheless, it can also be seen that this corpus is the only one where the automatic evaluation measures rank the systems exactly as the human judges do. On all other corpora, no automatic evaluation measure is able to rank the participating systems correctly (i.e., the way human judges do).

Table 8.3: Baseline and default parameters and methods for all experiments.

Parameter/Method	Baseline setting	Experimental setting	
		Sentence level	System level
Evaluator normalization	none	$(0,1)$-normalization	none
Case	ignore case	ignore case	use case
Punctuation	`mteval`	`mteval`; treat abbreviations	
Summation of scores	weighted	-	weighted
Reference length	average	best relative sentence	
Sentence boundaries	none	initial and end	
BLEU smoothing	none	BLEU-S	
Substitution cost	constant		
Evaluator aggregation	average		

Table 8.4: Effect of baseline settings and experimental default settings on the correlation with human evaluation. Pearson's r on sentence level.

Human score	Automatic measure + settings	TIDES 2002 CE	TIDES 2003 CE	TIDES 2003 AE	TIDES 2004 CE	TIDES 2004 AE	BTEC 2004 CE	BTEC 2004 JE
A	WER baseline	0.220	0.256	0.386	0.451	0.542	0.598	0.649
	default	0.320	0.349	0.505	0.540	0.597	0.691	**0.744**
	PER baseline	0.237	0.313	0.370	0.506	0.538	0.640	0.671
	default	0.329	0.428	0.495	0.579	0.600	**0.708**	**0.744**
	BLEU baseline	0.223	0.284	0.389	0.451	0.503	0.483	0.555
	default	0.404	0.451	0.541	**0.606**	**0.621**	0.570	0.635
	NIST baseline	0.388	0.435	0.492	0.563	0.565	0.512	0.577
	default	**0.434**	**0.513**	**0.562**	0.600	0.604	0.520	0.579
F	WER baseline	0.178	0.224	0.322	0.438	0.442	0.532	0.582
	default	0.277	0.301	0.423	0.511	0.496	**0.565**	**0.624**
	PER baseline	0.170	0.203	0.286	0.435	0.373	0.454	0.495
	default	0.245	0.298	0.389	0.493	0.424	0.456	0.504
	BLEU baseline	0.160	0.193	0.302	0.384	0.391	0.380	0.451
	default	**0.354**	**0.368**	**0.458**	**0.540**	**0.527**	0.390	0.462
	NIST baseline	0.280	0.246	0.372	0.428	0.395	0.275	0.339
	default	0.329	0.343	0.440	0.459	0.429	0.277	0.339
A+F	WER baseline	0.220	0.265	0.387	0.476	0.533	0.631	0.683
	default	0.328	0.365	0.518	0.559	0.589	**0.702**	**0.761**
	PER baseline	0.227	0.291	0.360	0.507	0.497	0.613	0.650
	default	0.321	0.419	0.496	0.575	0.556	0.653	0.697
	BLEU baseline	0.214	0.268	0.379	0.451	0.485	0.482	0.560
	default	0.416	0.464	0.556	**0.612**	**0.618**	0.539	0.612
	NIST baseline	0.372	0.388	0.476	0.537	0.524	0.443	0.513
	default	**0.427**	**0.498**	**0.563**	0.572	0.560	0.448	0.514

Table 8.5: Effect of baseline settings and experimental default settings on the correlation with A + F. Kendall's $\bar{\tau}$ on sentence level.

Automatic measure + settings	TIDES 2002 CE	TIDES 2003 CE	TIDES 2003 AE	TIDES 2004 CE	TIDES 2004 AE	BTEC 2004 CE	BTEC 2004 JE
WER baseline	0.076	0.126	0.276	0.303	0.290	0.390	0.559
default	0.145	0.193	0.372	0.363	0.317	0.389	**0.573**
PER baseline	0.119	0.173	0.284	0.350	0.291	0.376	0.535
default	0.185	0.271	0.366	0.382	0.317	0.364	0.534
BLEU baseline	0.121	0.183	0.290	0.322	0.313	**0.411**	0.537
default	0.230	0.286	0.389	**0.400**	**0.328**	0.262	0.463
NIST baseline	0.205	0.234	0.339	0.366	0.302	0.247	0.405
default	**0.235**	**0.309**	**0.397**	0.386	0.305	0.248	0.401

Table 8.6: Effect of baseline settings and experimental default settings on the correlation with A + F. Pearson's r on system level.

Automatic measure + settings	TIDES 2002 CE	TIDES 2003 CE	TIDES 2003 AE	TIDES 2004 CE	TIDES 2004 AE	BTEC 2004 CE	BTEC 2004 JE
WER baseline	-0.056	0.543	0.845	0.918	0.988	**0.909**	0.949
default	0.339	0.813	**0.928**	0.957	**0.994**	0.898	**0.979**
PER baseline	0.064	0.720	0.820	0.967	0.962	0.844	0.933
default	0.455	0.907	0.919	0.969	0.965	0.776	0.922
BLEU baseline	0.238	0.840	0.925	0.987	**0.993**	0.890	0.951
default	**0.618**	**0.927**	0.924	**0.989**	0.989	0.690	0.923
NIST baseline	0.436	0.828	0.917	0.952	0.971	0.480	0.782
default	0.530	0.907	0.915	0.956	0.985	0.429	0.766

Table 8.7: Effect of baseline settings and experimental default settings on the correlation with A + F. Kendall's τ on system level.

Automatic measure + settings	TIDES 2002 CE	TIDES 2003 CE	TIDES 2003 AE	TIDES 2004 CE	TIDES 2004 AE	BTEC 2004 CE	BTEC 2004 JE
WER baseline	0.056	0.333	0.600	0.733	**1.000**	0.745	**0.929**
default	0.167	**0.619**	**0.733**	0.822	**1.000**	**0.818**	**0.929**
PER baseline	0.000	0.524	0.467	0.911	0.800	0.636	0.714
default	0.278	**0.619**	**0.733**	0.822	0.800	0.636	0.714
BLEU baseline	0.278	**0.619**	**0.733**	**0.956**	**1.000**	0.782	0.857
default	**0.444**	**0.619**	**0.733**	0.867	**1.000**	0.564	0.786
NIST baseline	0.333	0.524	**0.733**	0.867	**1.000**	0.455	0.571
default	0.389	**0.619**	**0.733**	0.778	**1.000**	0.527	0.571

8.3 BLEU smoothing

In the next experiment, the effect of different BLEU smoothing methods, as described in Section 3.1.2, has been investigated. As Table 8.9 shows, smoothing improves the correlation with human evaluation for all corpora, although not with 95% confidence. Except for the BTEC 2004 CE corpus, the original smoothing method by [Lin & Och 04b] has a slightly higher correlation with human evaluation than our modified method BLEU-S'.

8.4 Tokenization and case normalization

Experiments on *tokenization* and the treatment of *punctuation marks* and *abbreviations* show, as can be seen in Table 8.9, that the most important step is *any* treatment of punctuation marks at all. Whether this step is the complete removal of punctuation marks, or their tokenization as "words" of their own may depend on the conditions. For the BTEC corpora, human evaluators were instructed to ignore punctuation at all; consequently some of the candidate corpora (but not the test corpora) were submitted without any punctuation marks. Consequently, the removal of punctuation achieves the highest correlation for these corpora. On the TIDES corpora for the NIST evaluations, no significant differences between the treatment of abbreviations plus the tokenization of punctuation marks on the one hand and the complete removal of punctuation marks can be found in terms of correlation with human evaluation, although the former has a slightly higher correlation in most cases. Using *case information* clearly has a negative effect on the correlation with human evaluation, at least on sentence level. Whereas Table 8.9 lists only the experimental for WER, this effect is similar for the other automatic evaluation measures.

8.5 Reference length calculation

The experimental results regarding different *Reference Length* calculation methods are surprising in so far as this much-neglected subject has a higher influence on correlation with human evaluation than might be expected: Just by changing from "average reference length" to "length of relative-best reference" increases the correlation between WER and

Table 8.8: Effect of BLEU smoothing methods on the correlation between BLEU and A + F. Pearson's r on sentence level.

BLEU smoothing	TIDES 2002 CE	TIDES 2003 CE	TIDES 2003 AE	TIDES 2004 CE	TIDES 2004 AE	BTEC 2004 CE	BTEC 2004 JE
no smoothing	0.386	0.438	0.540	0.599	0.591	0.612	0.686
95%-Conf.	*±0.019*	*±0.020*	*±0.022*	*±0.019*	*±0.031*	*±0.017*	*±0.017*
BLEU-S	**0.403**	**0.452**	**0.548**	**0.614**	**0.603**	0.632	**0.702**
BLEU-S'	0.396	0.446	0.544	0.604	0.595	**0.636**	0.700

Table 8.9: Effect of different tokenization and case normalization steps on the correlation between WER and A + F. Pearson's r on sentence level.

Tokenization method	TIDES 2002 CE	TIDES 2003 CE	TIDES 2003 AE	TIDES 2004 CE	TIDES 2004 AE	BTEC 2004 CE	BTEC 2004 JE
keep punctuation	0.299	0.356	0.425	0.486	0.583	0.691	0.748
95%-Conf.	*±0.020*	*±0.021*	*±0.025*	*±0.022*	*±0.030*	*±0.014*	*±0.013*
remove punctuation	0.318	0.361	0.499	0.551	**0.599**	**0.726**	**0.771**
tokenize punctuation	0.320	**0.367**	0.480	**0.561**	0.589	0.690	0.748
+ treat abbrev.	**0.328**	0.365	**0.518**	0.559	0.589	0.702	0.761
+ abbrev. + use case	0.306	0.349	0.493	0.554	0.535	0.613	0.664

human evaluation significantly, as shown in Table 8.10. On almost all corpora, this increase goes well beyond 95% confidence. The effects on PER are similar.

Taking the average length instead of the minimum or the closest length seems to be the best choice for both BLEU (Table 8.11) and the NIST measure (Table 8.12), at least for the TIDES corpora. For the BTEC corpora with their many reference sentences, both closest length and minimum length perform significantly better in terms of correlation with human evaluation.

8.6 m-gram-based distance measures

Table 8.13 shows the correlation between several m-gram count-vector–based distance measures, as introduced in Section 4.2, and adequacy or fluency. In this table, MSDER denotes the *multiset distance* between the unigram count vectors. As can be seen, the corresponding error measure has a lower or even significantly lover correlation with human evaluation than PER has for almost all corpora Whereas the unigram PER corresponds better to adequacy than the bigram PER, the latter clearly has the higher correlation of them with fluency. For the BTEC corpora, *skip bigrams* with a moderate maximum skip seem to perform better than regular bigrams, but this effect cannot be confirmed on the

Table 8.10: Effect of different reference length calculation methods on the correlation between WER and A + F. Pearson's r on sentence level.

Reference length method	TIDES 2002 CE	TIDES 2003 CE	TIDES 2003 AE	TIDES 2004 CE	TIDES 2004 AE	BTEC 2004 CE	BTEC 2004 JE
avg length	0.251	0.311	0.482	0.496	0.568	0.645	0.694
95%-Conf.	*±0.021*	*±0.022*	*±0.023*	*±0.022*	*±0.031*	*±0.015*	*±0.016*
avg nearest	0.278	0.344	0.496	0.529	0.582	0.663	0.730
best	**0.328**	**0.365**	**0.518**	**0.559**	**0.589**	**0.702**	**0.761**

Table 8.11: Effect of different reference length calculation methods on the correlation between BLEU and A + F. Pearson's r on sentence level.

Reference length method	TIDES 2002 CE	TIDES 2003 CE	TIDES 2003 AE	TIDES 2004 CE	TIDES 2004 AE	BTEC 2004 CE	BTEC 2004 JE
avg length	**0.375**	**0.435**	**0.527**	0.591	**0.594**	0.613	0.651
95%-Conf.	*±0.019*	*±0.020*	*±0.022*	*±0.019*	*±0.030*	*±0.016*	*±0.018*
min length	0.360	0.423	0.518	**0.593**	0.581	**0.680**	**0.724**
min nearest	0.358	0.422	0.519	**0.593**	0.581	**0.679**	**0.724**

Table 8.12: Effect of different reference length calculation methods on the correlation between NIST and A + F. Pearson's r on sentence level.

Reference length method	TIDES 2002 CE	TIDES 2003 CE	TIDES 2003 AE	TIDES 2004 CE	TIDES 2004 AE	BTEC 2004 CE	BTEC 2004 JE
avg length	**0.427**	**0.498**	**0.563**	0.572	**0.560**	0.448	0.514
95%-Conf.	*±0.018*	*±0.018*	*±0.021*	*±0.019*	*±0.031*	*±0.021*	*±0.022*
min length	0.408	0.492	0.555	**0.595**	0.545	**0.602**	**0.663**
min nearest	0.407	0.492	0.555	**0.595**	0.545	**0.602**	**0.663**

TIDES corpora. For all corpora, trigrams and higher m-grams (see also Table B.12) perform even worse than bigrams in terms of correlation; the same holds for combinations of several m-gram distances.

8.7 Sentence boundaries

As can be seen in Table 8.14, the use of *Sentence Boundaries* increases the correlation between bigram PER and human evaluation significantly. For BLEU (Table 8.15), the same holds, although for only the TIDES corpora. For the NIST measure (Table 8.16), no such effect can be noticed.

8.8 Block movement distance measures

A smaller subset consisting of candidate sentences shorter than 20 words for each MT system allowed for experiments not only including INVWER, but also LJWER. The surprising result (Table 8.17) is that CDER has by far the highest correlation with human evaluation for all TIDES corpora, even though it does not penalize redundant or superfluous candidate words. For the BTEC corpora, both LJWER and INVWER have a higher correlation with human evaluation than the CDER. Nevertheless, CDER still has a higher correlation than BLEU on all corpora but one.

Table 8.13: Correlation of PER, m-PER, and skip-bigram PER with human evaluation. Pearson's r on sentence level.

Human score	Count vector on	TIDES 2002 CE	TIDES 2003 CE	TIDES 2003 AE	TIDES 2004 CE	TIDES 2004 AE	BTEC 2004 CE	BTEC 2004 JE
A	Unigram MSDER	0.316	0.384	**0.505**	0.508	0.559	0.700	0.717
	Unigram PER	**0.329**	**0.428**	0.495	**0.579**	0.600	0.708	0.744
	95%-Conf.	*±0.020*	*±0.020*	*±0.023*	*±0.019*	*±0.029*	*±0.013*	*±0.014*
	Bigram Skip=0	0.294	0.374	0.475	0.554	0.585	0.657	0.704
	Skip=4	0.284	0.359	0.478	0.529	**0.606**	0.717	**0.757**
	Skip=10	0.262	0.349	0.455	0.515	0.601	**0.723**	**0.757**
	Trigram	0.243	0.306	0.427	0.495	0.532	0.594	0.641
	(1...4)-gram	0.288	0.367	0.469	0.551	0.579	0.650	0.693
F	Unigram MSDER	0.246	0.292	**0.407**	0.456	0.409	0.536	0.561
	Unigram PER	0.245	0.298	0.389	0.493	0.424	0.456	0.504
	95%-Conf.	*±0.021*	*±0.022*	*±0.026*	*±0.022*	*±0.038*	*±0.021*	*±0.023*
	Bigram Skip=0	**0.248**	**0.306**	**0.407**	**0.513**	**0.485**	0.549	0.591
	Skip=4	0.236	0.294	0.402	0.488	**0.485**	**0.579**	**0.618**
	Skip=10	0.210	0.286	0.384	0.472	0.479	0.558	0.595
	Trigram	0.215	0.274	0.383	0.482	0.467	0.544	0.585
	(1...4)-gram	0.236	0.298	0.401	0.512	0.477	0.527	0.578

Table 8.14: Effect of sentence boundaries on the correlation between BIGRAM-PER and A + F. Pearson's r on sentence level.

Sentence boundaries	TIDES 2002 CE	TIDES 2003 CE	TIDES 2003 AE	TIDES 2004 CE	TIDES 2004 AE	BTEC 2004 CE	BTEC 2004 JE
no boundaries	0.258	0.355	0.465	0.540	0.570	0.666	0.698
95%-Conf.	*±0.020*	*±0.021*	*±0.024*	*±0.020*	*±0.031*	*±0.014*	*±0.016*
+ initial	0.276	0.368	0.478	0.555	0.575	**0.690**	**0.733**
+ end	0.283	0.375	0.482	0.558	0.569	0.640	0.677
+ both	**0.299**	**0.384**	**0.491**	**0.568**	**0.577**	0.674	0.720

Table 8.15: Effect of sentence boundaries on the correlation between BLEU and A + F. Pearson's r on sentence level.

Sentence boundaries	TIDES 2002 CE	TIDES 2003 CE	TIDES 2003 AE	TIDES 2004 CE	TIDES 2004 AE	BTEC 2004 CE	BTEC 2004 JE
no boundaries	0.358	0.422	0.519	0.593	0.581	**0.679**	**0.724**
95%-Conf.	*±0.019*	*±0.020*	*±0.023*	*±0.019*	*±0.032*	*±0.014*	*±0.015*
+ initial	0.367	0.430	0.527	0.604	0.593	0.659	0.721
+ end	0.400	0.447	0.541	0.605	0.590	0.632	0.687
+ both	**0.403**	**0.452**	**0.548**	**0.614**	**0.603**	0.632	0.702

Table 8.16: Effect of sentence boundaries on the correlation between NIST and A + F. Pearson's r on sentence level.

Sentence boundaries	TIDES 2002 CE	TIDES 2003 CE	TIDES 2003 AE	TIDES 2004 CE	TIDES 2004 AE	BTEC 2004 CE	BTEC 2004 JE
no boundaries	0.427	**0.498**	0.563	**0.572**	0.560	**0.448**	**0.514**
95%-Conf.	*±0.018*	*±0.019*	*±0.022*	*±0.020*	*±0.033*	*±0.021*	*±0.023*
+ initial	0.431	0.494	0.558	0.567	**0.577**	0.333	0.417
+ end	0.429	0.497	**0.565**	0.569	0.558	0.403	0.472
+ both	**0.432**	0.493	0.561	0.560	0.570	0.290	0.373

Table 8.17: Correlation of different block move distances with A + F on 20 word-corpora. Pearson's r on sentence level.

Automatic evaluation measure	TIDES 2002 CE	TIDES 2003 CE	TIDES 2003 AE	TIDES 2004 CE	TIDES 2004 AE	BTEC 2004 CE	BTEC 2004 JE
BLEU	0.495	0.477	**0.612**	0.599	0.636	0.622	0.698
95%-Conf.	*±0.037*	*±0.042*	*±0.040*	*±0.047*	*±0.057*	*±0.016*	*±0.016*
WER	0.402	0.400	0.546	0.567	0.637	0.693	**0.757**
CDER	**0.527**	**0.482**	0.609	**0.628**	**0.666**	0.669	0.719
LJWER	0.405	0.422	0.564	0.586	0.639	0.697	0.749
INVWER	0.409	0.426	0.578	0.599	0.631	**0.699**	0.754

8.9 $\overline{CD}$CD-distance, CDER

To confirm the assumption that a *complete* and *disjunct* coverage (CD) of the reference sentence is of a higher importance for the correctness of a candidate sentence than a *complete* and *disjunct* coverage of the candidate itself, an experiment was conducted where both directions of $d_{\overline{CD}CD}$ were taken as evaluation measure. Moreover, both the maximum and the sum of these two directions were taken as additional measures. Table 8.18 shows that CDER correlates not only far more with human evaluation if it is calculated in the direction postulated in Section 4.4.2 rather than in the opposed direction, but even both sum and maximum both have a significantly lower correlation with human evaluation than the original direction has.

Whether or not sentence boundaries are important for CDER; that is, whether or not an additional long jump is necessary if the first word of the reference sentence is not aligned to the first word in the candidate sentence, or the last reference word not to the last candidate word – is of little importance, as Table 8.19 reveals: The differences between the correlation is hardly relevant on any corpus.

8.10 $\overline{CD}$CD-distance with miscoverage penalty

This leaves a problem for the original CDER, namely with "babbling" MT systems. An MT system can be called "babbling" if it tends to put any even slightly probable word and/or phrase into the candidate translation to achieve the best possible *recall*. With the best possible *recall*, a low CDER will be achieved, too. Training parameters on a test set using CDER would thus favor these "babbling" systems. This would be prevented with the sum or maximum of both possible CDER directions, but as has been showed in the previous experiment, doing this decreases the correlation of the measure and human evaluation significantly. Consequently, a penalty for overly long or repetitive candidate sentences should be considered. Unfortunately, Table 8.20 reveals that both miscoverage penalties proposed in Section 4.4.3 as well as their average do effect a significant decrease in the correlation between CDER and human evaluation on the TIDES corpora.

Table 8.18: Effect of different application directions on the correlation between CDER and A + F. Pearson's r on sentence level.

Direction	TIDES 2002 CE	TIDES 2003 CE	TIDES 2003 AE	TIDES 2004 CE	TIDES 2004 AE	BTEC 2004 CE	BTEC 2004 JE
CD for candidate	**0.424**	**0.458**	**0.544**	**0.625**	**0.623**	0.680	0.724
95%-Conf.	*±0.018*	*±0.019*	*±0.022*	*±0.018*	*±0.028*	*±0.014*	*±0.014*
CD for reference	0.157	0.171	0.422	0.222	0.393	0.548	0.565
sum of both	0.305	0.343	0.521	0.482	0.563	**0.722**	0.747
maximum of both	0.337	0.402	0.523	0.594	0.599	0.702	**0.749**

Table 8.19: Effect of "Boundaries" on the correlation between CDER and A + F. Pearson's r on sentence level.

sentence boundaries	TIDES 2002 CE	TIDES 2003 CE	TIDES 2003 AE	TIDES 2004 CE	TIDES 2004 AE	BTEC 2004 CE	BTEC 2004 JE
no boundaries	0.419	0.455	0.554	0.623	0.611	**0.703**	**0.758**
95%-Conf.	*±0.018*	*±0.019*	*±0.021*	*±0.018*	*±0.029*	*±0.013*	*±0.013*
+ initial	**0.427**	**0.462**	**0.555**	**0.626**	0.621	0.701	0.743
+ end	0.420	0.452	0.545	0.623	0.615	0.684	0.741
+ both	0.424	0.458	0.544	0.625	**0.623**	0.680	0.724

Table 8.20: Effect of different miscoverage penalty functions on the correlation between CDER and A + F. Pearson's r on sentence level.

CDER miscoverage penalty	TIDES 2002 CE	TIDES 2003 CE	TIDES 2003 AE	TIDES 2004 CE	TIDES 2004 AE	BTEC 2004 CE	BTEC 2004 JE
CDER	**0.424**	**0.458**	**0.544**	**0.625**	**0.623**	0.680	0.724
95%-Conf.	*±0.018*	*±0.019*	*±0.022*	*±0.018*	*±0.028*	*±0.014*	*±0.014*
+ path miscoverage	0.277	0.309	0.469	0.466	0.528	0.677	0.721
+ length difference	0.300	0.377	0.473	0.581	0.567	0.698	**0.746**
+ $\frac{1}{2}$ both	0.299	0.349	0.483	0.534	0.557	**0.700**	0.741

8.11 Word-dependent substitution costs

The outcome of the experiments on word-dependent substitution costs are listed in Table 8.21: For WER, both prefix-dependent substitution costs and Levenshtein-dependent substitution costs (see Section 4.5) increase the correlation with human evaluation significantly for the TIDES corpora. For the BTEC corpus, the decrease in correlation is measurable, but not significant. Both cost schemes perform similar; a clear favorite cannot be determined. For PER, the increase in correlation when using prefix-dependent substitution costs is remarkably lower than the increase for WER. With Levenshtein-dependent substitution costs, even a decrease in correlation can be noticed. The probable explanation for this is that it is too easy for the algorithm to find a "sufficiently similar", but completely unrelated reference word to match a wrong candidate word.

CDER benefits from word-dependent substitution costs as well. On the TIDES corpora, Levenshtein substitution costs have a slightly higher correlation than prefix ones. On the other hand, they have a significantly lower correlation with human evaluation on the BTEC corpora. Consequently, prefix-dependent substitution costs are the method of choice for CDER, as well.

8.12 Linear combination of evaluation measures

Weighted linear combination of different evaluation measures is a wide field of research; a full exploration would have been gone beyond the scope of this study. Especially the need for cross validation and similar methods to prevent training on test data for the weight vectors would have required appropriate consideration. Nevertheless, preliminary proof of concept experiments have been conducted. Figure 8.1 shows how a weighted linear combination of CDER and PER, both using prefix-dependent substitution costs, can increase the correlation between the combined measure and adequacy. For all corpora, a combination of 60 percent CDER and 40 percent PER has a significantly higher correlation with adequacy than the original measures.

8.13 Overview: Before and after this thesis

Table 8.22 gives a final overview of how the different evaluation measures and methods of this work can be used to increase the correlation between automatic and human evaluation on sentence level. For fairness, the "baseline" experiments in this table included evaluator and case normalization as well. Starting with BLEU and WER, the application of improved settings, namely treatment of abbreviations, a better reference length determination method, smoothing, and sentence boundaries can augment the correlation with human evaluation. An additional increase in correlation can be achieved using CDER instead of BLEU. More improvement in correlation comes from the use of prefix-dependent substitution costs for CDER. Finally, a linear combination of CDER and PER as described in Section 8.12 gives an additional increase in the correlation with the sum of adequacy and fluency for the TIDES corpora.

Table 8.21: Effect of word-dependent substitution costs on the correlation between automatic evaluation and A + F. Pearson's r on sentence level.

Measure with c_{SUB} depending on	TIDES 2002 CE	TIDES 2003 CE	TIDES 2003 AE	TIDES 2004 CE	TIDES 2004 AE	BTEC 2004 CE	BTEC 2004 JE
WER	0.328	0.365	0.518	0.559	0.589	**0.702**	**0.761**
95%-Conf.	*±0.020*	*±0.021*	*±0.022*	*±0.020*	*±0.030*	*±0.013*	*±0.013*
+ prefix	0.356	0.389	0.530	0.571	0.605	0.695	0.759
+ Levenshtein	0.354	0.388	0.531	0.580	0.611	0.681	0.750
PER	0.321	0.419	0.496	0.575	0.556	0.653	0.697
95%-Conf.	*±0.020*	*±0.020*	*±0.023*	*±0.019*	*±0.032*	*±0.015*	*±0.016*
+ prefix	0.345	0.450	0.507	0.577	0.556	0.622	0.668
+ Levenshtein	0.320	0.435	0.489	0.553	0.542	0.578	0.634
CDER	0.424	0.458	0.544	0.625	0.623	0.680	0.724
95%-Conf.	*±0.018*	*±0.019*	*±0.022*	*±0.018*	*±0.028*	*±0.014*	*±0.014*
+ prefix	0.453	0.484	0.555	0.637	0.634	0.672	0.722
+ Levenshtein	**0.457**	**0.484**	**0.559**	**0.638**	**0.637**	0.658	0.710

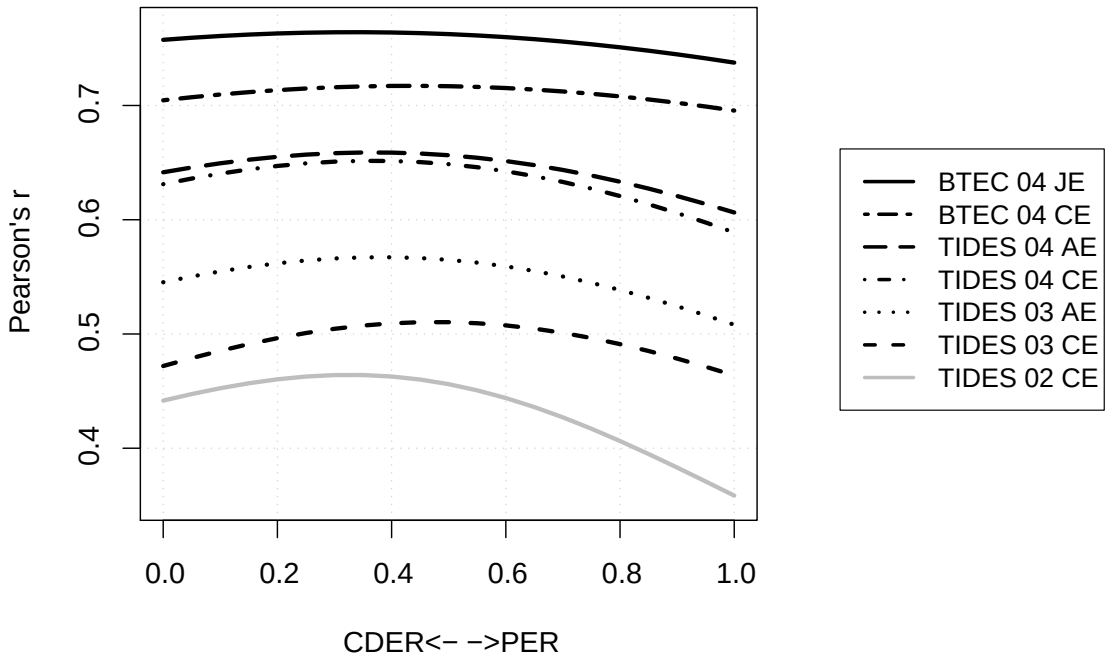


Figure 8.1: Effects of the weights on the correlation between adequacy and a weighted linear combination of CDER and PER. Pearson's r on sentence level.

Table 8.22: Effect of this work on the correlation between automatic and human evaluation. Pearson's r on sentence level.

Human score	Settings and measure	TIDES 2002 CE	TIDES 2003 CE	TIDES 2003 AE	TIDES 2004 CE	TIDES 2004 AE	BTEC 2004 CE	BTEC 2004 JE
A	WER baseline	0.244	0.292	0.419	0.480	0.572	0.598	0.649
	95%-Conf.	*±0.021*	*±0.023*	*±0.026*	*±0.023*	*±0.033*	*±0.017*	*±0.018*
	BLEU baseline	0.250	0.327	0.436	0.484	0.530	0.483	0.555
	+ default*	0.404	0.451	0.541	0.606	0.621	0.570	0.635
	CDER + def.†	0.411	0.449	0.535	0.615	0.630	0.703	0.750
	+ w-c_{SUB}‡	0.442	0.472	0.545	0.631	0.642	0.705	0.757
	+ PER§	**0.460**	**0.510**	**0.567**	**0.651**	**0.659**	**0.717**	**0.764**
F	WER baseline	0.216	0.272	0.349	0.466	0.479	**0.532**	**0.581**
	95%-Conf.	*±0.021*	*±0.023*	*±0.028*	*±0.023*	*±0.037*	*±0.019*	*±0.021*
	BLEU baseline	0.201	0.257	0.347	0.409	0.419	0.380	0.451
	+ default	0.354	0.368	**0.458**	0.540	0.527	0.390	0.462
	CDER + def.	0.363	0.357	0.440	0.557	0.525	0.511	0.550
	+ w-c_{SUB}	**0.383**	**0.381**	0.449	**0.560**	**0.535**	0.495	0.537
	+ PER	0.370	**0.381**	0.456	**0.559**	0.515	0.473	0.515
A+F	WER baseline	0.251	0.313	0.429	0.502	0.566	0.631	0.683
	95%-Conf.	*±0.021*	*±0.022*	*±0.026*	*±0.022*	*±0.033*	*±0.016*	*±0.017*
	BLEU baseline	0.251	0.333	0.440	0.477	0.513	0.482	0.560
	+ default	0.416	0.464	0.556	0.612	0.618	0.539	0.612
	CDER + def.	0.424	0.458	0.544	0.625	0.623	**0.680**	**0.724**
	+ w-c_{SUB}	0.453	0.484	0.555	0.637	**0.634**	0.672	0.722
	+ PER	**0.460**	**0.511**	**0.573**	**0.649**	**0.635**	0.667	0.714

*BLEU with default settings instead of baseline settings. See Section 8.2.

†CDER with default settings.

‡CDER with default settings and prefix-length dependent substitution costs. See Section 8.11.

§Weighted combination of 60% CDER and 40% PER. Default settings and prefix-dependent substitution costs. See Section 8.12.

Chapter 9

Conclusion and Perspectives

In this chapter, conclusions are drawn from the experimental results in this work. Related topics to this work are listed that, to the author's opinion, require further investigation.

Conclusion

In this work, the automatic evaluation of Machine Translation using reference-based evaluation measures has been studied. The study has portrayed a rich set of different similarity and error measures for automatic MT evaluation. First, two well established similarity measures have been described, namely the BLEU similarity measure and the NIST measure. Afterwards, the family of distance-based error measures has been introduced. All members of this family are based on a string distance function, which is normalized by a reference length into an error measure. Within these distance measures, the first subclass of measures is based on the comparison of the m-gram count vectors of candidate and reference sentence. Starting from the well-established PER measure, an extension of PER to arbitrary m-grams and skip bigrams have been introduced. Furthermore, a multiset-based count vector distance has been defined. Comparing m-grams instead of only unigrams has the advantage that the ordering of the words within a sentence has an influence on the score. Anyhow, experimental benchmarks have shown that the bigram PER for fluency and the unigram PER for adequacy are superior to all other m-gram-based distance measures in correlation. Alone a 10-skip bigram PER showed a higher correlation, but only on a few test sets.

Then, string-based distance measures have been introduced in this work. Starting from the well-known Levenshtein distance, which is the basis for the WER measure, several new extensions have been introduced. These extensions are based on the requirement of allowing reordering in the sentence, because this is a common procedure in MT. The central challenge is that finding an optimal (i.e., cost-minimal) set of reordering operations is an NP-hard problem. Approximate solutions are hardly useable in MT evaluation, as has been explained in this study. The first approach to overcome this difficulty has been to restrict the possible set of permutations, either by limiting the number of block transpositions – and thus making it a fixed-parameter–tractable problem – or by allowing only specific sets of reordering operations. The latter is the idea of the INVWER distance

measure, which is based on bracketing transduction grammars. Consequently, the INVWER distance can be calculated in bicubic and thus polynomial time. The presentation of a Dynamic Programming algorithm and an optimized Memoization algorithm in this work has proved this time complexity constructively.

The second approach to simplify the block reordering problem that has been investigated in this study is to relax the restrictions on alignments between candidate and reference words, namely to drop the constraints that both sentences have to be covered completely and disjunctively. This simplification has lead to a modification of the Levenshtein algorithm. The new algorithm solves this search problem in quadratic time. Since the new measure is no longer symmetric, a specification is necessary whether the distance is calculated in the direction from candidate sentence to reference sentence or vice versa. This study has stated that the more suitable way for MT evaluation is to require complete and disjunct coverage of the reference sentence only. Later experimental results have clearly confirmed this statement. Unfortunately, an MT evaluation using only this CDER measure would favor "babbling" MT systems, which are systems that tend to produce overly long candidate translations containing each possibly correct word. Although the investigated corpora have not contained such candidate translations, a later use of the pure CDER measure for evaluation or even training purposes can surely be expected to provoke them. Consequently, methods to penalize overly long candidate translations have been presented. Regrettably, all presented methods have lowered the correlation of the measure with human evaluation in experiments significantly.

Word-dependent substitution costs have been shown to further improve the edit-distance–based measures, namely WER, PER, and CDER. The idea is that words with a rather similar spelling often have a similar meaning. Therefore, substituting words with a similar meaning should not be penalized as harshly as substituting words with a completely different meaning. Two methods for word-dependent substitution cost functions have been presented, namely a character-based Levenshtein distance between the substituted words, and a function based on the common prefix length of the words. Experimental results have shown that WER and especially CDER gain from this extension significantly in their correlation with human evaluation. PER, where additionally a more sophisticated algorithm is required, gains only marginally from it.

Altogether, the new evaluation measure CDER constantly has shown a higher correlation with human evaluation than the best-established evaluation measure, BLEU. On some corpora, other measures such as NIST or WER have exhibited an even higher correlation, but as this prevalence has been rather unstable, CDER can be considered as the "best" measure for general purpose.

The presentation of the different evaluation measures themselves in this work has been followed by a presentation of preprocessing steps and auxiliary methods common to most implementations of automatic evaluation measures. The first presented step has been the tokenization of the input sentence, in particular the treatment of punctuation marks and abbreviations, and the use or non–use of case information. Experimental results have shown the importance of implementing a special treatment of punctuation marks. Whether the optimal treatment is the complete removal of these marks, or their tokenization (i.e., their treatment as separate words), depends on the test conditions and the corpus. The inclusion of case information has experimentally been shown to lower

the correlation between automatic and human evaluation on the analyzed test corpora, whereas a normalization of abbreviations has been proven useful.

Different methods of determining the reference length for a candidate sentence have been the next item addressed in this study. All presented automatic evaluation measures require such a method. Experiments in this study have confirmed that the new method for distance-based error measures that has been defined in this work achieves the highest correlation between human evaluation and these measures. For both NIST and BLEU, the results on the TIDES test corpora indicate that the default method for NIST correlates with human evaluation to a slightly higher degree than the default method for BLEU. On the BTEC corpora, the default method for BLEU has exhibited the higher correlation, with a significantly larger difference in the correlations. An explanation for this might be that the latter test corpora contain four times more reference translations than the former, with a more heterogeneous length distribution.

A new feature for automatic evaluation measures has been the inclusion of sentence boundaries for m-gram-based evaluation measures. Experiments have proved that this extension increases the correlation between human evaluation and BLEU or m-PER significantly.

On the side of human MT evaluation, steps must be taken to prevent or cancel out evaluator bias. Whereas there are simple steps to do this for system evaluation, sentence evaluation requires additional investigation. Experiments have shown that $(0,1)$-normalization for the human assessments can augment inter-annotator correlation as well as they increase the correlation between human and automatic evaluation.

All methods and measures that have been presented in this study have been evaluated by comparing the correlation among the automatic evaluation measure and human evaluation measures, namely fluency and adequacy. Basis for these experiments have been seven international MT evaluation campaigns from the previous three years. Five of them had been in the domain of news articles and similar texts, two had been in the tourism and personal communication domain. The source language in most of these campaigns had been Chinese; additionally, Arabic and Japanese had been used. The target language in all campaigns had been English. Because even in the largest campaign no more than eleven systems had been evaluated under comparable conditions consistent correlation measurements with a reasonable confidence range could not have been expected. Consequently, all correlation coefficients have been calculated on sentence level, rather than on system level.

Perspectives

With the high correlation between human evaluation and a rather simple measure, the CDER, the question arises why this is the case – especially with regard to CDER not penalizing superfluous and even completely wrong words in candidate sentences. All attempts to introduce at least a length penalty effected in a significantly lower correlation with human evaluation. Without any penalization of "babbling" MT systems, the CDER alone is of little worth for the comparison of MT systems, and completely worthless for MT parameter optimization. This effect nourishes the suspicion that human evaluators

might tend to judge unnecessary repetitions and additions of words more harshly than omissions of relevant facts. With this in mind, further investigations should take place, first on whether this is really the case, and second on how this could be modelled in an automatic evaluation measure; for example, the deletion costs c_{DEL} could be adjusted accordingly. After all, if longer and more redundant sentences are what human users want, this is what MT systems should produce.

The hunt for the "perfect" automatic evaluation measure is not over yet. With structurally rather unlike evaluation measures on the pole positions of correlation, namely the m-gram precision BLEU and the string edit distance CDER, there are still lots of possibilities for measures between these two, or even completely different from them. Although the full long jump distance showed a lower correlation than CDER in the experiments, improvements on the deletion problem could possibly change the inferiority of INVWER, too. With the calculation of this distance being fixed-parameter–tractable in the number of long jumps, further algorithmic improvements could push forward this measure again. Naive approaches have not born fruit, because it needs at least five independent long jumps for LJWER to handle more permutations than INVWER. For the approaches considered during the study, this amount of long jumps still yields a complexity too high for practical use.

Whereas fluency and adequacy as human evaluation measures are evaluated independently in most evaluation campaigns, no such distinction is made for automatic evaluation measures. Each such measure is considered to judge the overall quality of a candidate sentence or system, rather than the quality with respect to certain aspects. However, the experiments for this work have shown that some measures have preferences for certain aspects – the unigram PER correlates with adequacy to a higher degree than the bigram PER, whereas this is vice versa on the fluency. Thus it should be investigated whether such preferences can be exploited for a more detailed evaluation.

Furthermore, the success of word-dependent substitution costs in improving evaluation measures proves that fixed costs are not the last word on the subject. Neither will probably be the presented functions; the syntactical and the semantical difference should actually play a major rôle in these costs, just as the actual importance of the words. How this can be done should be the focus of further research.

To enable the researcher to decide between significant improvements of his or her MT system on the one side and random effects on the other side, the output of an automatic MT evaluation tool should include confidence estimations. One possibility for this would be a confidence range, giving an estimation on how much an independent system must be better in terms of the evaluation measure to be considered the superior system with a certain confidence. Another possibility would be a direct, sentence-wise comparison of two MT systems using the **bootstrapping** mechanism, with the null hypothesis that both systems are equal. The latter approach is already used in speech recognition research [Bisani & Ney 04]. With this implemented, the system ranking capabilities of the different automatic evaluation measures will have to be reestimated, as a system that is neither significantly better nor significantly worse than another system in terms of the measure should be ranked equally to the other system.

An area that definitely needs further research is the weighted combination of different evaluation measures, especially as these parameters can easily be optimized. Although

it could be considered "cheating" if these weights would be optimized on the training data, machine learning research has developed techniques to handle similar cases for years. Among these techniques, *cross validation* schemes or even *Leaving One Out* should be considered.

A possible field of application for this combined measure could be larger evaluation campaigns. Here, human judges can evaluate a representative (!) part of the candidate sentences at first. Then, weight vectors and other evaluation parameters can be adjusted for optimal correlation using these samples. The optimized measure can then be used for the evaluation of the whole candidate sets.

Appendix A

Software documentation

In this appendix, a short introduction to the `EvalTrans` evaluation software will be given. With this software, all evaluation experiments for this study have been conducted.

A.1 `EvalTransBatchEval`

All experiments for this work were conducted using the now freely available `EvalTrans` software package [Leusch & Nießen+ 02]. This framework for human and automatic evaluation of machine translation is written in Tcl/Tk [Ousterhout & TclCoreTeam 04]. To achieve an acceptable runtime as well as a higher reusability, each automatic evaluation measure is implemented in C++ and linked to the main software using the `Swig` wrapper generator [Beazley & Fulton+ 02].

For batch mode evaluation, the `EvalTrans` package contains the `EvalTransBatchEval` script, which is called as follows:

```
EvalTransBatchEval.tcl \
    --database reference database file; see Section A.2          \
    --source   source sentences file; one sentence per line      \
    --target   candidate sentences file; one sentence per line   \
    [additional options and flags ...]
```

For the additional options, see Tables A.1 to A.8. Note that for compatibility reasons, the *default* setting for these options is not necessarily the *default* setting for the experiments in this paper. Therefore, these options should be set explicitly for experiments. For example, sentence boundaries are only used for BLEU if `--bleu-pad-rule` is set accordingly.

A.2 `EvalTrans` database file format

An commented example of the `Xml` database format for `EvalTrans` can be found in Figure A.1. This format is capable of storing reference translations as well as evaluated

Table A.1: Options for `EvalTransBatchEval`.

Option	Description
`--output-file`	Output report file
`--encoding`	Input/Output encoding; e.g. `iso8859-1` or `utf-8`
`--comment`	Comment for report
`--wer-ref-length-rule`	Reference length rule for WER, PER, etc – see Table A.3.
`--cder-pad-rule`	Sentence boundaries for CDER– see Table A.6.
`--cder-coverage-rule`	Miscoverage penalty for CDER– see Table A.7.
`--nper-max-ngram`	Maximum m-gram size for m-PER (4)
`--nper-pad-rule`	Sentence boundaries for m-PER– see Table A.5.
`--nper-ref-length-rule`	Reference length rule for m-PER– see Table A.3.
`--max-ngram`	Maximum m-gram size for BLEU (4)
`--bleu-pad-rule`	Sentence boundaries for BLEU– see Table A.5.
`--bleu-ref-length-rule`	Reference length rule for BLEU– see Table A.4.
`--nist-max-ngram`	Maximum m-gram size for NIST (5)
`--nist-pad-rule`	Sentence boundaries for NIST– see Table A.5.
`--nist-ref-length-rule`	Reference length rule for NIST– see Table A.4.
`--substitution-cost-rule`	Rule for calculating word dependent substitution costs. WER, PER and CDER only – see Table A.8.
`--transposition-costs`	Costs in d_{inv}, as comma separated list of integers: *denominator*,c_{SUB},c_{DEL},c_{INS},c_{INV}.
`--max-bigram-skip`	Maximum bigram skip for Skip-bigram PER.
`--leaving-one-out`	Ignore identical reference sentences for the listed measures. List of measures, e.g. `BLEU,mWER`.

Table A.2: Flags for `EvalTransBatchEval`.

Flag	Description
`--extrapolate`	Estimate human evaluation from database.
`--nper-choose-references-separately`	For m-PER, minimize separately for each m.
`--all-ngrams`	Calculate m-gram scores for each single M.
`--sentences-in-detail`	Show scores and distances for each single candidate sentence.
`--include-sentences`	Include the actual candidate and reference sentences in the report.
`--allow-transpositions`	Calculate INVWER.
`--allow-long-jumps`	Calculate LJWER (approximatively for sentences > 25 words).
`--xml-mode`	Generate output file in XML format. Mandatory for the calculation of anything exceeding BLEU and WER– see Section A.3.

Table A.3: Reference length determination methods for WER, PER, etc.

Value	Description
`average`	Average reference length
`minimum`	Minimum reference length
`maximum`	Maximum reference length
`averageNearest`	Average length of reference sentences with lowest absolute distance
`minimumNearest`	Minimum length of reference sentences with lowest absolute distance
`maximumNearest`	Maximum length of reference sentences with lowest absolute distance
`lowestError`	Length of the reference sentence with lowest relative distance

Table A.4: Reference length determination methods for BLEU and NIST.

Value	Description
`average`	Average reference length
`minimum`	Minimum reference length
`minimumNearest`	Minimum length of reference sentences with lowest length difference

Table A.5: Sentence boundaries for BIGRAM-PER, BLEU and NIST.

Value	Description
`none`	No boundaries
`left`	Sentence initial
`right`	Sentence end
`both`	Both sentence initial and end

Table A.6: Sentence boundaries for CDER.

Value	Description
`none`	Sentence initial and end arbitrary
`left`	Sentence initial fixed, end arbitrary
`right`	Sentence initial arbitrary, end fixed
`both`	Sentence initial and end fixed

Table A.7: Miscoverage penalties CDER.

Value	Description
`none`	No penalty
`length`	Length difference
`miscoverage`	Absolute miscoverage
`half`	Average of length difference and absolute miscoverage

Table A.8: Word dependent substitution costs.

Value	Description
`none`	Fixed substitution costs only
`prefix`	Prefix length substitution costs
`levenshtein`	Levenshtein substitution costs

candidate translations. Therefore, several different sentence scores can be stored. The last defined score in the database is taken into account for the selection of reference sentence. Each target sentence where this score is larger or equal to a previously defined "reference" level is considered a reference sentence. For this, it can be helpful to define a special `isReference` score, as in the example. Note that the sentences in a test set are differentiated by their actual source sentences only, rather than by their position. Consequently it is not possible to have a source sentence with different sets of reference translations. If this is necessary, an artificial set of source sentences containing an ID for each position must be created.

A.3 `EvalTrans` Report file format

The `EvalTrans` XML report file format was developed to contain all possible evaluation measures both on system and (if requested) on sentence level, as well as all necessary parameters and settings to reproduce an evaluation. Easy downward-compatible extensibility and a simple parsing were additional goals. For this, a hierarchical XML format was defined. Disadvantage of this approach is that the report files can be rather large, i.e. several 100,000 lines. Nevertheless, commonly available XML tools, such as XSL processors allow for a simple analysis of the results. A commented overview of a report file can be found in Figure A.2.

```
<?xml version="1.0" encoding="utf-8" ?>
<!DOCTYPE etdb SYSTEM "etdb.dtd">
<database>
  <version_id>RCS version ID</version_id>
  <!-- Human evaluation score definition: -->
  <levels class="Name" count="# Levels (incl. 0)"
          gray="GUI use only"
          green="GUI use only"
          perfect="Reference level"
          firstvalid="Lowest valid level"/>
  <!-- E.g.: -->
  <levels class="fluency-min" count="6" gray="2" green="5"
                              perfect="5" firstvalid="1"/>
  ...
  <!-- The last listed score here is relevant for the
       selection of reference sentences: -->
  <levels class="isPerfect" count="2" gray="0" green="1"
                            perfect="1" firstvalid="0"/>

  <source doc="Optional document ID; not used by EvalTrans"
          seg="Optional segment ID; not used by EvalTrans">
    <s_sent>First source sentence</s_sent>
    <targets>
      <tgt sys="Optional MT System ID; not used by EvalTrans">
        <t_sent>First target sentence.
                Reference or candidate sentence, depending on last score.</t_sent>
        <!-- Scores, such as: -->
        <eval class="fluency" val="3"/>
        ...
        <eval class="isPerfect" val="1"/>
      </tgt>
      <!-- More targets ... -->
    </targets>
  </source>
  <!-- More sources ... -->
</database>
```

Figure A.1: The `EvalTrans` database file format.

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE etreport SYSTEM "etreport.dtd">
<report>
  <author>User name</author>
  <date>Date of evaluation</date>
  <comment>...</comment>
  <software><!--Software revision information--></software>
  <files>
    <database-file>Database path</database-file>
    <source-file>Source path</source-file>
    ...
    <command-line>EvalTransBatchEval command line</command-line>
  </files>
  <candidate-statistics>
    <!--Some statistics about the candidate sentences-->
  </candidate-statistics>

  <evaluation-options>
    <!--Options and flags for varying evaluation measures:-->
    <wer-options>
      <reference-length-rule>lowestError</reference-length-rule>
    </wer-options>
    <cder-options>
      <reference-length-rule>lowestError</reference-length-rule>
      <coverage-rule>half</coverage-rule>
      <pad-rule>both</pad-rule>
    </cder-options>
    ...
  </evaluation-options>

  <evaluation><!--System evaluation -- see figure A.3--></evaluation>

  <candidate-evaluation>
    <candidate nr="0">
      <source nr="0"><!--Source sentence information--></source>
      <target nr="-1"><!--Candidate and reference sentence information--></target>
      <!--Sentence evaluation analogously to figure A.3 with additional
          <reference-distance/> elements for absolute distances.          -->
    </candidate>
    ...
  </candidate-evaluation>
</report>
```

Figure A.2: The `EvalTrans` report file format.

```
<wer>
  <error-rate>WER </error-rate>
  <error-rate-percent>WER in percent</error-rate-percent>
  <reference-length>Total reference length</reference-length>
  <sum>Total distance</sum>
</wer>
<wwer> <!--Analogously: WER with word dependent substitution costs--></wwer>
<cder>
  <forward>
    <error-rate>CDER with CD for reference sentence</error-rate>
    ...
  </forward>
  <backward>Analogously: CDER with CD for candidate sentence</backward>
  <bidirectional>Analogously: average CDER for both directions</bidirectional>
  <maximum>Analogously: maximum CDER for both directions</maximum>
 </cder>
 ...
 <nper>
   <!--(1... M)-PER -->
   <n-gram n="1"><!--Unigram PER --></n-gram>
   ...
 </nper>
<msder><!--Analogously: MSDER --></msder>
<wper><!--Analogously: PER with word dependent substitution costs analogously--></wper>
 ...
<bleu>
  <score-unpenalized>BLEU score without length penalty</score-unpenalized>
  <score>BLEU score with length penalty</score>
  <s-score-unpenalized>BLEU-S score without length penalty</score-unpenalized> ...
  <r-score-unpenalized>BLEU-S' score without length penalty</score-unpenalized> ...
  <length-penalty>BLEU length penalty</length-penalty>
  <n-gram n="1"><!--Unigram part of BLEU score--></n-gram>
  ...
</bleu>
<nist><!--Analogously to BLEU: NIST score--></nist>
...
```

Figure A.3: The `EvalTrans` report file format: The system evaluation part.

Appendix B

Additional results

Additional results that would only have cluttered chapter 8 can be found in this appendix. How the tables are to be read can be found in said chapter.

Table B.1: Effect of baseline settings and experimental default settings on the correlation with human evaluation. Kendall's $\bar{\tau}$ on sentence level.

Human score	Automatic measure + settings	TIDES 2002 CE	TIDES 2003 CE	TIDES 2003 AE	TIDES 2004 CE	TIDES 2004 AE	BTEC 2004 CE	BTEC 2004 JE
A	WER baseline	0.073	0.131	0.291	0.299	0.294	0.378	0.554
	default	0.141	0.200	0.374	0.351	0.298	0.393	0.584
	PER baseline	0.117	0.202	0.296	0.360	0.293	**0.424**	0.587
	default	0.180	0.276	0.372	0.380	0.315	**0.424**	**0.603**
	BLEU baseline	0.115	0.210	0.298	0.325	0.306	0.377	0.532
	default	0.220	0.296	0.386	**0.399**	**0.325**	0.300	0.510
	NIST baseline	0.212	0.282	0.355	0.387	0.312	0.311	0.492
	default	**0.233**	**0.320**	**0.405**	0.396	0.305	0.310	0.487
F	WER baseline	0.070	0.106	0.228	0.279	0.250	0.310	**0.463**
	default	0.131	0.135	0.284	0.321	0.266	0.312	0.459
	PER baseline	0.099	0.114	0.238	0.304	0.242	0.272	0.408
	default	0.158	0.188	0.282	0.313	**0.272**	0.261	0.392
	BLEU baseline	0.106	0.129	0.249	0.284	0.267	**0.366**	0.455
	default	**0.208**	0.197	0.306	**0.325**	0.271	0.207	0.355
	NIST baseline	0.165	0.145	0.281	0.296	0.248	0.178	0.280
	default	0.193	**0.206**	**0.314**	0.303	0.245	0.179	0.276
A+F	WER baseline	0.076	0.126	0.276	0.303	0.290	0.390	0.559
	default	0.145	0.193	0.372	0.363	0.317	0.389	**0.573**
	PER baseline	0.119	0.173	0.284	0.350	0.291	0.376	0.535
	default	0.185	0.271	0.366	0.382	0.317	0.364	0.534
	BLEU baseline	0.121	0.183	0.290	0.322	0.313	**0.411**	0.537
	default	0.230	0.286	0.389	**0.400**	**0.328**	0.262	0.463
	NIST baseline	0.205	0.234	0.339	0.366	0.302	0.247	0.405
	default	**0.235**	**0.309**	**0.397**	0.386	0.305	0.248	0.401

Table B.2: Effect of baseline settings and experimental default settings on the correlation with human evaluation. Pearson's r on system level.

Human score	Automatic measure + settings	TIDES 2002 CE	TIDES 2003 CE	TIDES 2003 AE	TIDES 2004 CE	TIDES 2004 AE	BTEC 2004 CE	BTEC 2004 JE
A	WER baseline	-0.059	0.516	0.840	0.893	0.971	0.293	0.820
	default	0.301	0.809	**0.931**	0.945	0.989	0.384	0.905
	PER baseline	0.042	0.729	0.806	0.965	0.982	0.594	0.900
	default	0.412	0.912	0.920	0.974	0.979	**0.635**	**0.946**
	BLEU baseline	0.177	0.823	0.921	0.976	0.979	0.256	0.864
	default	**0.544**	0.916	0.923	**0.987**	0.981	0.417	0.921
	NIST baseline	0.386	0.876	0.914	0.969	0.988	0.511	0.928
	default	0.474	**0.925**	0.916	0.973	**0.994**	0.550	0.919
F	WER baseline	-0.048	0.529	0.841	0.936	**0.998**	**0.912**	**0.939**
	default	0.394	0.789	0.910	0.960	0.994	0.864	0.897
	PER baseline	0.097	0.600	0.830	0.953	0.924	0.722	0.806
	default	0.515	0.840	0.905	0.944	0.936	0.630	0.728
	BLEU baseline	0.326	0.761	**0.916**	**0.985**	**0.999**	0.905	0.889
	default	**0.724**	**0.899**	0.914	0.975	0.993	0.617	0.760
	NIST baseline	0.492	0.607	0.908	0.910	0.937	0.343	0.462
	default	0.609	0.807	0.900	0.914	0.965	0.272	0.440
A+F	WER baseline	-0.056	0.543	0.845	0.918	0.988	**0.909**	0.949
	default	0.339	0.813	**0.928**	0.957	**0.994**	0.898	**0.979**
	PER baseline	0.064	0.720	0.820	0.967	0.962	0.844	0.933
	default	0.455	0.907	0.919	0.969	0.965	0.776	0.922
	BLEU baseline	0.238	0.840	0.925	0.987	**0.993**	0.890	0.951
	default	**0.618**	**0.927**	0.924	**0.989**	0.989	0.690	0.923
	NIST baseline	0.436	0.828	0.917	0.952	0.971	0.480	0.782
	default	0.530	0.907	0.915	0.956	0.985	0.429	0.766

Table B.3: Effect of baseline settings and experimental default settings on the correlation with human evaluation. Kendall's τ on system level.

Human score	Automatic measure + settings	TIDES 2002 CE	TIDES 2003 CE	TIDES 2003 AE	TIDES 2004 CE	TIDES 2004 AE	BTEC 2004 CE	BTEC 2004 JE
A	WER baseline	-0.056	0.429	0.600	0.689	**1.000**	0.055	**0.929**
	default	0.111	**0.619**	**0.733**	0.733	**1.000**	0.273	**0.929**
	PER baseline	-0.111	**0.619**	0.467	0.867	0.800	**0.382**	0.714
	default	0.222	**0.619**	**0.733**	0.911	0.800	**0.382**	0.857
	BLEU baseline	0.167	0.524	**0.733**	0.911	**1.000**	0.091	0.857
	default	**0.389**	**0.619**	**0.733**	**0.956**	**1.000**	0.164	0.786
	NIST baseline	0.222	**0.619**	**0.733**	0.911	**1.000**	0.200	0.714
	default	0.333	**0.619**	**0.733**	0.867	**1.000**	0.273	0.714
F	WER baseline	0.111	0.238	0.600	0.778	**1.000**	0.855	**0.714**
	default	0.278	0.524	**0.733**	0.822	**1.000**	0.782	**0.714**
	PER baseline	0.056	0.429	0.467	0.867	0.800	0.673	0.500
	default	0.389	0.524	**0.733**	0.733	0.800	0.527	0.500
	BLEU baseline	0.333	0.524	**0.733**	**0.911**	**1.000**	**0.891**	0.643
	default	**0.556**	**0.714**	**0.733**	0.778	**1.000**	0.527	0.571
	NIST baseline	0.389	0.429	**0.733**	0.733	**1.000**	0.345	0.357
	default	0.500	0.524	**0.733**	0.689	**1.000**	0.418	0.357
A+F	WER baseline	0.056	0.333	0.600	0.733	**1.000**	0.745	**0.929**
	default	0.167	**0.619**	**0.733**	0.822	**1.000**	**0.818**	**0.929**
	PER baseline	0.000	0.524	0.467	0.911	0.800	0.636	0.714
	default	0.278	**0.619**	**0.733**	0.822	0.800	0.636	0.714
	BLEU baseline	0.278	**0.619**	**0.733**	**0.956**	**1.000**	0.782	0.857
	default	**0.444**	**0.619**	**0.733**	0.867	**1.000**	0.564	0.786
	NIST baseline	0.333	0.524	**0.733**	0.867	**1.000**	0.455	0.571
	default	0.389	**0.619**	**0.733**	0.778	**1.000**	0.527	0.571

Table B.4: Effect of BLEU smoothing methods on the correlation between BLEU and human evaluation. Pearson's r on sentence level.

Human score	BLEU smoothing	TIDES 2002 CE	TIDES 2003 CE	TIDES 2003 AE	TIDES 2004 CE	TIDES 2004 AE	BTEC 2004 CE	BTEC 2004 JE
A	no smoothing	0.377	0.427	0.527	0.591	0.600	0.608	0.671
	95%-Conf.	*±0.019*	*±0.020*	*±0.023*	*±0.019*	*±0.031*	*±0.017*	*±0.017*
	BLEU-S	**0.387**	**0.436**	**0.531**	**0.603**	**0.603**	**0.637**	**0.697**
	BLEU-S′	0.384	0.433	0.529	0.596	0.601	**0.636**	0.691
F	no smoothing	0.323	0.345	0.442	0.531	0.498	0.486	0.564
	95%-Conf.	*±0.020*	*±0.022*	*±0.025*	*±0.021*	*±0.036*	*±0.020*	*±0.022*
	BLEU-S	**0.348**	**0.365**	**0.456**	**0.549**	**0.516**	0.493	0.565
	BLEU-S′	0.336	0.356	0.447	0.537	0.503	**0.501**	**0.568**
A+F	no smoothing	0.386	0.438	0.540	0.599	0.591	0.612	0.686
	95%-Conf.	*±0.019*	*±0.020*	*±0.022*	*±0.019*	*±0.031*	*±0.017*	*±0.017*
	BLEU-S	**0.403**	**0.452**	**0.548**	**0.614**	**0.603**	0.632	**0.702**
	BLEU-S′	0.396	0.446	0.544	0.604	0.595	**0.636**	0.700

Table B.5: Effect of different tokenization and case normalization steps on the correlation between WER and human evaluation. Pearson's r on sentence level.

Human score	Tokenization method	TIDES 2002 CE	TIDES 2003 CE	TIDES 2003 AE	TIDES 2004 CE	TIDES 2004 AE	BTEC 2004 CE	BTEC 2004 JE
A	keep punctuation	0.287	0.330	0.406	0.466	0.581	0.677	0.731
	95%-Conf.	*±0.020*	*±0.022*	*±0.026*	*±0.023*	*±0.030*	*±0.014*	*±0.014*
	remove punctuation	0.309	0.345	0.486	0.535	**0.611**	**0.690**	0.733
	tokenize punctuation	0.310	**0.349**	0.472	**0.541**	0.594	0.677	0.731
	+ treat abbrev.	**0.320**	**0.349**	**0.505**	0.540	0.597	**0.691**	**0.744**
	+ abbrev. + use case	0.305	0.332	0.485	0.532	0.547	0.616	0.664
F	keep punctuation	0.263	**0.314**	0.355	0.451	**0.503**	0.559	0.615
	95%-Conf.	*±0.020*	*±0.022*	*±0.027*	*±0.023*	*±0.034*	*±0.018*	*±0.019*
	remove punctuation	0.273	0.301	0.405	0.501	0.499	**0.611**	**0.656**
	tokenize punctuation	0.274	0.308	0.386	**0.513**	0.500	0.558	0.614
	+ treat abbrev.	**0.277**	0.301	**0.423**	0.511	0.496	0.565	0.624
	+ abbrev. + use case	0.248	0.289	0.395	0.510	0.445	0.480	0.528
A+F	keep punctuation	0.299	0.356	0.425	0.486	0.583	0.691	0.748
	95%-Conf.	*±0.020*	*±0.021*	*±0.025*	*±0.022*	*±0.030*	*±0.014*	*±0.013*
	remove punctuation	0.318	0.361	0.499	0.551	**0.599**	**0.726**	**0.771**
	tokenize punctuation	0.320	**0.367**	0.480	**0.561**	0.589	0.690	0.748
	+ treat abbrev.	**0.328**	0.365	**0.518**	0.559	0.589	0.702	0.761
	+ abbrev. + use case	0.306	0.349	0.493	0.554	0.535	0.613	0.664

Table B.6: Effect of different tokenization and case normalization steps on the correlation between PER and human evaluation. Pearson's r on sentence level.

Human score	Tokenization method	TIDES 2002 CE	TIDES 2003 CE	TIDES 2003 AE	TIDES 2004 CE	TIDES 2004 AE	BTEC 2004 CE	BTEC 2004 JE
A	keep punctuation	0.284	0.388	0.375	0.513	0.553	0.698	0.737
	95%-Conf.	*±0.020*	*±0.021*	*±0.026*	*±0.021*	*±0.032*	*±0.013*	*±0.014*
	remove punctuation	0.321	**0.432**	0.485	**0.580**	**0.602**	**0.722**	**0.747**
	tokenize punctuation	0.322	0.423	0.472	**0.579**	**0.602**	0.696	0.736
	+ treat abbrev.	**0.329**	0.428	**0.495**	**0.579**	0.600	0.708	0.744
	+ abbrev. + use case	0.316	0.373	0.443	0.562	0.536	0.588	0.625
F	keep punctuation	0.230	**0.304**	0.316	0.447	0.413	0.454	0.501
	95%-Conf.	*±0.021*	*±0.022*	*±0.028*	*±0.023*	*±0.038*	*±0.021*	*±0.023*
	remove punctuation	0.242	**0.304**	0.380	0.486	0.420	**0.535**	**0.565**
	tokenize punctuation	**0.246**	0.296	0.364	**0.492**	**0.430**	0.452	0.499
	+ treat abbrev.	0.245	0.298	**0.389**	**0.493**	0.424	0.456	0.504
	+ abbrev. + use case	0.217	0.261	0.331	0.487	0.365	0.325	0.375
A+F	keep punctuation	0.284	0.394	0.387	0.514	0.523	0.646	0.691
	95%-Conf.	*±0.020*	*±0.020*	*±0.026*	*±0.021*	*±0.033*	*±0.015*	*±0.016*
	remove punctuation	0.315	**0.424**	0.486	0.572	0.556	**0.703**	**0.731**
	tokenize punctuation	0.317	0.416	0.470	**0.575**	**0.560**	0.644	0.690
	+ treat abbrev.	**0.321**	0.419	**0.496**	**0.575**	0.556	0.653	0.697
	+ abbrev. + use case	0.300	0.368	0.438	0.562	0.490	0.513	0.559

Table B.7: Effect of different tokenization and case normalization steps on the correlation between BLEU and human evaluation. Pearson's r on sentence level.

Human score	Tokenization method	TIDES 2002 CE	TIDES 2003 CE	TIDES 2003 AE	TIDES 2004 CE	TIDES 2004 AE	BTEC 2004 CE	BTEC 2004 JE
A	keep punctuation	0.311	0.374	0.461	0.545	0.558	0.646	0.697
	95%-Conf.	*±0.020*	*±0.021*	*±0.024*	*±0.020*	*±0.032*	*±0.015*	*±0.016*
	remove punctuation	0.323	0.395	0.488	0.580	0.577	**0.661**	**0.711**
	tokenize punctuation	0.333	0.398	0.485	0.578	0.575	0.646	0.696
	+ treat abbrev.	0.387	**0.436**	**0.531**	**0.603**	**0.603**	0.637	0.697
	+ abbrev. + use case	**0.397**	0.413	0.507	0.592	0.567	0.534	0.582
F	keep punctuation	0.320	0.357	0.427	0.521	0.513	0.558	0.595
	95%-Conf.	*±0.020*	*±0.021*	*±0.025*	*±0.021*	*±0.034*	*±0.018*	*±0.020*
	remove punctuation	0.311	0.347	0.430	0.531	0.503	**0.614**	**0.634**
	tokenize punctuation	0.315	0.344	0.421	0.531	0.501	0.556	0.594
	+ treat abbrev.	0.348	**0.365**	**0.456**	**0.549**	**0.516**	0.493	0.565
	+ abbrev. + use case	**0.350**	0.354	0.426	**0.549**	0.480	0.361	0.418
A+F	keep punctuation	0.340	0.405	0.490	0.567	0.574	0.672	0.718
	95%-Conf.	*±0.019*	*±0.020*	*±0.023*	*±0.020*	*±0.031*	*±0.014*	*±0.015*
	remove punctuation	0.344	0.416	0.509	0.592	0.581	**0.711**	**0.747**
	tokenize punctuation	0.353	0.417	0.503	0.591	0.579	0.671	0.717
	+ treat abbrev.	0.403	**0.452**	**0.548**	**0.614**	**0.603**	0.632	0.702
	+ abbrev. + use case	**0.410**	0.433	0.521	0.608	0.563	0.502	0.557

Table B.8: Effect of different reference length calculation methods on the correlation between WER and human evaluation. Pearson's r on sentence level.

Human score	Reference length method	TIDES 2002 CE	TIDES 2003 CE	TIDES 2003 AE	TIDES 2004 CE	TIDES 2004 AE	BTEC 2004 CE	BTEC 2004 JE
A	avg length	0.245	0.293	0.466	0.476	0.577	0.612	0.661
	95%-Conf.	*±0.021*	*±0.022*	*±0.024*	*±0.022*	*±0.031*	*±0.016*	*±0.017*
	min length	0.204	0.286	0.279	0.463	0.568	0.562	0.619
	max length	0.249	0.272	0.467	0.452	0.556	0.571	0.636
	avg nearest	0.273	0.329	0.487	0.512	0.595	0.660	0.719
	min nearest	0.252	0.321	0.475	0.494	0.585	0.610	0.673
	max nearest	0.277	0.327	0.488	0.518	**0.596**	0.669	0.731
	best	**0.320**	**0.349**	**0.505**	**0.540**	**0.597**	**0.691**	**0.744**
F	avg length	0.213	0.267	0.400	0.458	0.477	0.544	0.587
	95%-Conf.	*±0.021*	*±0.023*	*±0.026*	*±0.023*	*±0.036*	*±0.018*	*±0.020*
	min length	0.157	0.258	0.234	0.439	0.446	0.454	0.513
	max length	0.224	0.252	0.410	0.446	0.476	0.548	0.597
	avg nearest	0.234	0.285	0.397	0.482	0.484	0.526	0.593
	min nearest	0.208	0.276	0.384	0.464	0.469	0.480	0.547
	max nearest	0.245	0.286	0.400	0.490	0.490	0.532	0.607
	best	**0.277**	**0.301**	**0.423**	**0.511**	**0.496**	**0.565**	**0.624**
A+F	avg length	0.251	0.311	0.482	0.496	0.568	0.645	0.694
	95%-Conf.	*±0.021*	*±0.022*	*±0.023*	*±0.022*	*±0.031*	*±0.015*	*±0.016*
	min length	0.200	0.302	0.287	0.480	0.548	0.568	0.630
	max length	0.258	0.290	0.487	0.477	0.556	0.624	0.685
	avg nearest	0.278	0.344	0.496	0.529	0.582	0.663	0.730
	min nearest	0.254	0.335	0.482	0.509	0.569	0.609	0.679
	max nearest	0.286	0.343	0.497	0.536	0.586	0.671	0.744
	best	**0.328**	**0.365**	**0.518**	**0.559**	**0.589**	**0.702**	**0.761**

Table B.9: Effect of different reference length calculation methods on the correlation between PER and human evaluation. Pearson's r on sentence level.

Human score	Reference length method	TIDES 2002 CE	TIDES 2003 CE	TIDES 2003 AE	TIDES 2004 CE	TIDES 2004 AE	BTEC 2004 CE	BTEC 2004 JE
A	avg length	0.259	0.370	0.443	0.532	0.572	0.653	0.688
	95%-Conf.	*±0.020*	*±0.021*	*±0.025*	*±0.021*	*±0.031*	*±0.015*	*±0.016*
	min length	0.225	0.361	0.296	0.518	0.563	0.595	0.639
	max length	0.267	0.357	0.450	0.524	0.559	0.616	0.674
	avg nearest	0.307	0.417	0.482	0.563	0.594	0.687	0.719
	min nearest	0.290	0.401	0.462	0.549	0.580	0.648	0.682
	max nearest	0.316	0.424	0.490	0.571	**0.600**	0.696	0.728
	best	**0.329**	**0.428**	**0.495**	**0.579**	**0.600**	**0.708**	**0.744**
F	avg length	0.192	0.267	0.357	0.458	0.397	0.463	0.503
	95%-Conf.	*±0.021*	*±0.023*	*±0.027*	*±0.023*	*±0.039*	*±0.021*	*±0.023*
	min length	0.147	0.260	0.236	0.441	0.370	0.379	0.431
	max length	0.205	0.259	0.371	0.458	0.401	**0.477**	**0.522**
	avg nearest	0.223	0.286	0.372	0.477	0.413	0.429	0.481
	min nearest	0.206	0.273	0.353	0.463	0.396	0.402	0.455
	max nearest	0.233	0.294	0.381	0.485	0.423	0.432	0.487
	best	**0.245**	**0.298**	**0.389**	**0.493**	**0.424**	0.456	0.504
A+F	avg length	0.252	0.366	0.448	0.531	0.527	0.625	0.665
	95%-Conf.	*±0.021*	*±0.021*	*±0.024*	*±0.021*	*±0.033*	*±0.016*	*±0.017*
	min length	0.210	0.356	0.298	0.514	0.509	0.546	0.597
	max length	0.263	0.354	0.458	0.526	0.521	0.611	0.666
	avg nearest	0.297	0.407	0.481	0.558	0.547	0.626	0.671
	min nearest	0.279	0.391	0.459	0.543	0.532	0.590	0.635
	max nearest	0.307	0.415	0.490	0.567	**0.556**	0.633	0.679
	best	**0.321**	**0.419**	**0.496**	**0.575**	**0.556**	**0.653**	**0.697**

Table B.10: Effect of different reference length calculation methods on the correlation between BLEU and human evaluation. Pearson's r on sentence level.

Human score	Reference length method	TIDES 2002 CE	TIDES 2003 CE	TIDES 2003 AE	TIDES 2004 CE	TIDES 2004 AE	BTEC 2004 CE	BTEC 2004 JE
A	avg length	**0.359**	**0.421**	**0.510**	**0.585**	**0.595**	0.623	0.664
	95%-Conf.	*±0.019*	*±0.020*	*±0.023*	*±0.019*	*±0.030*	*±0.016*	*±0.017*
	min length	0.342	0.406	0.499	0.583	0.580	**0.655**	**0.704**
	min nearest	0.340	0.405	0.500	0.583	0.580	**0.655**	**0.704**
F	avg length	**0.326**	**0.349**	**0.437**	0.521	**0.509**	0.473	0.505
	95%-Conf.	*±0.020*	*±0.021*	*±0.025*	*±0.021*	*±0.034*	*±0.020*	*±0.023*
	min length	0.318	0.346	0.434	**0.529**	0.500	**0.563**	**0.599**
	min nearest	0.316	0.345	0.435	**0.530**	0.499	**0.562**	**0.599**
A+F	avg length	**0.375**	**0.435**	**0.527**	0.591	**0.594**	0.613	0.651
	95%-Conf.	*±0.019*	*±0.020*	*±0.022*	*±0.019*	*±0.030*	*±0.016*	*±0.018*
	min length	0.360	0.423	0.518	**0.593**	0.581	**0.680**	**0.724**
	min nearest	0.358	0.422	0.519	**0.593**	0.581	**0.679**	**0.724**

Table B.11: Effect of different reference length calculation methods on the correlation between NIST and human evaluation. Pearson's r on sentence level.

Human score	Reference length method	TIDES 2002 CE	TIDES 2003 CE	TIDES 2003 AE	TIDES 2004 CE	TIDES 2004 AE	BTEC 2004 CE	BTEC 2004 JE
A	avg length	**0.434**	**0.513**	**0.562**	0.600	**0.604**	0.520	0.579
	95%-Conf.	*±0.018*	*±0.018*	*±0.021*	*±0.018*	*±0.029*	*±0.019*	*±0.020*
	min length	0.412	0.502	0.551	**0.615**	0.587	**0.656**	**0.703**
	min nearest	0.411	0.502	0.550	**0.615**	0.586	**0.655**	**0.703**
F	avg length	**0.329**	0.343	**0.440**	0.459	**0.429**	0.277	0.339
	95%-Conf.	*±0.020*	*±0.021*	*±0.025*	*±0.023*	*±0.038*	*±0.024*	*±0.027*
	min length	0.319	**0.348**	**0.440**	**0.490**	0.420	**0.418**	**0.485**
	min nearest	0.317	**0.348**	**0.440**	**0.490**	0.419	**0.417**	**0.485**
A+F	avg length	**0.427**	**0.498**	**0.563**	0.572	**0.560**	0.448	0.514
	95%-Conf.	*±0.018*	*±0.018*	*±0.021*	*±0.019*	*±0.031*	*±0.021*	*±0.022*
	min length	0.408	0.492	0.555	**0.595**	0.545	**0.602**	**0.663**
	min nearest	0.407	0.492	0.555	**0.595**	0.545	**0.602**	**0.663**

Table B.12: Correlation of PER, m-PER, and skip-bigram PER with human evaluation. Pearson's r on sentence level.

Human score	Count vector on	TIDES 2002 CE	TIDES 2003 CE	TIDES 2003 AE	TIDES 2004 CE	TIDES 2004 AE	BTEC 2004 CE	BTEC 2004 JE
A	Unigram MSDER	0.316	0.384	**0.505**	0.508	0.559	0.700	0.717
	Unigram PER	**0.329**	**0.428**	0.495	**0.579**	0.600	0.708	0.744
	95%-Conf.	*±0.020*	*±0.020*	*±0.023*	*±0.019*	*±0.029*	*±0.013*	*±0.014*
	Bigram Skip=0	0.294	0.374	0.475	0.554	0.585	0.657	0.704
	Skip=4	0.284	0.359	0.478	0.529	**0.606**	0.717	**0.757**
	Skip=10	0.262	0.349	0.455	0.515	0.601	**0.723**	**0.757**
	Skip=25	0.229	0.331	0.425	0.481	0.586	**0.722**	0.755
	Trigram	0.243	0.306	0.427	0.495	0.532	0.594	0.641
	4-gram	0.212	0.264	0.396	0.446	0.482	0.545	0.599
	(1…4)-gram	0.288	0.367	0.469	0.551	0.579	0.650	0.693
F	Unigram MSDER	0.246	0.292	**0.407**	0.456	0.409	0.536	0.561
	Unigram PER	0.245	0.298	0.389	0.493	0.424	0.456	0.504
	95%-Conf.	*±0.021*	*±0.022*	*±0.026*	*±0.022*	*±0.038*	*±0.021*	*±0.023*
	Bigram Skip=0	**0.248**	**0.306**	**0.407**	**0.513**	**0.485**	0.549	0.591
	Skip=4	0.236	0.294	0.402	0.488	**0.485**	**0.579**	**0.618**
	Skip=10	0.210	0.286	0.384	0.472	0.479	0.558	0.595
	Skip=25	0.180	0.262	0.356	0.431	0.452	0.555	0.592
	Trigram	0.215	0.274	0.383	0.482	0.467	0.544	0.585
	4-gram	0.191	0.252	0.364	0.450	0.440	0.520	0.571
	(1…4)-gram	0.236	0.298	0.401	0.512	0.477	0.527	0.578
A+F	Unigram MSDER	0.313	0.385	**0.511**	0.514	0.525	0.692	0.712
	Unigram PER	**0.321**	**0.419**	0.496	**0.575**	0.556	0.653	0.697
	95%-Conf.	*±0.020*	*±0.020*	*±0.023*	*±0.019*	*±0.032*	*±0.015*	*±0.016*
	Bigram Skip=0	0.299	0.384	0.491	0.568	0.577	0.674	0.720
	Skip=4	0.286	0.369	0.490	0.542	**0.589**	**0.724**	**0.765**
	Skip=10	0.261	0.359	0.467	0.527	0.584	0.716	0.753
	Skip=25	0.226	0.337	0.436	0.487	0.561	0.715	0.750
	Trigram	0.251	0.323	0.449	0.518	0.538	0.635	0.680
	4-gram	0.219	0.285	0.420	0.474	0.495	0.593	0.649
	(1…4)-gram	0.289	0.376	0.484	0.567	0.570	0.658	0.707

Table B.13: Effect of sentence boundaries on the correlation between BIGRAM-PER and human evaluation. Pearson's r on sentence level.

Human score	Sentence boundaries	TIDES 2002 CE	TIDES 2003 CE	TIDES 2003 AE	TIDES 2004 CE	TIDES 2004 AE	BTEC 2004 CE	BTEC 2004 JE
A	no boundaries	0.256	0.348	0.451	0.532	0.587	0.646	0.691
	95%-Conf.	*±0.020*	*±0.021*	*±0.024*	*±0.021*	*±0.030*	*±0.015*	*±0.016*
	+ initial	0.273	0.359	0.463	0.544	**0.590**	**0.665**	**0.710**
	+ end	0.278	0.367	0.467	0.546	0.578	0.631	0.677
	+ both	**0.294**	**0.374**	**0.475**	**0.554**	0.585	0.657	0.704
F	no boundaries	0.212	0.278	0.382	0.479	0.469	0.546	0.565
	95%-Conf.	*±0.021*	*±0.022*	*±0.026*	*±0.022*	*±0.036*	*±0.018*	*±0.021*
	+ initial	0.227	0.291	0.394	0.498	0.475	**0.571**	**0.607**
	+ end	0.236	0.295	0.398	0.500	0.478	0.514	0.540
	+ both	**0.248**	**0.306**	**0.407**	**0.513**	**0.485**	0.549	0.591
A+F	no boundaries	0.258	0.355	0.465	0.540	0.570	0.666	0.698
	95%-Conf.	*±0.020*	*±0.021*	*±0.024*	*±0.020*	*±0.031*	*±0.014*	*±0.016*
	+ initial	0.276	0.368	0.478	0.555	0.575	**0.690**	**0.733**
	+ end	0.283	0.375	0.482	0.558	0.569	0.640	0.677
	+ both	**0.299**	**0.384**	**0.491**	**0.568**	**0.577**	0.674	0.720

Table B.14: Effect of sentence boundaries on the correlation between BLEU and human evaluation. Pearson's r on sentence level.

Human score	Sentence boundaries	TIDES 2002 CE	TIDES 2003 CE	TIDES 2003 AE	TIDES 2004 CE	TIDES 2004 AE	BTEC 2004 CE	BTEC 2004 JE
A	no boundaries	0.340	0.405	0.500	0.583	0.580	**0.655**	0.704
	95%-Conf.	*±0.020*	*±0.021*	*±0.024*	*±0.020*	*±0.032*	*±0.015*	*±0.016*
	+ initial	0.352	0.413	0.510	0.591	0.595	0.649	**0.704**
	+ end	0.382	0.430	0.523	0.594	0.587	0.640	0.692
	+ both	**0.387**	**0.436**	**0.531**	**0.603**	**0.603**	0.637	0.697
F	no boundaries	0.316	0.345	0.435	0.530	0.499	**0.562**	**0.599**
	95%-Conf.	*±0.020*	*±0.022*	*±0.026*	*±0.021*	*±0.036*	*±0.018*	*±0.020*
	+ initial	0.318	0.350	0.439	0.542	0.505	0.530	0.592
	+ end	**0.350**	0.362	0.452	0.539	0.511	0.490	0.542
	+ both	0.348	**0.365**	**0.456**	**0.549**	**0.516**	0.493	0.565
A+F	no boundaries	0.358	0.422	0.519	0.593	0.581	**0.679**	**0.724**
	95%-Conf.	*±0.019*	*±0.020*	*±0.023*	*±0.019*	*±0.032*	*±0.014*	*±0.015*
	+ initial	0.367	0.430	0.527	0.604	0.593	0.659	0.721
	+ end	0.400	0.447	0.541	0.605	0.590	0.632	0.687
	+ both	**0.403**	**0.452**	**0.548**	**0.614**	**0.603**	0.632	0.702

Table B.15: Effect of sentence boundaries on the correlation between NIST and human evaluation. Pearson's r on sentence level.

Human score	Sentence boundaries	TIDES 2002 CE	TIDES 2003 CE	TIDES 2003 AE	TIDES 2004 CE	TIDES 2004 AE	BTEC 2004 CE	BTEC 2004 JE
A	no boundaries	0.434	0.513	0.562	**0.600**	0.604	**0.520**	**0.579**
	95%-Conf.	*±0.018*	*±0.018*	*±0.022*	*±0.019*	*±0.031*	*±0.020*	*±0.021*
	+ initial	0.435	0.506	0.557	0.594	**0.617**	0.413	0.491
	+ end	0.436	**0.513**	**0.564**	0.598	0.604	0.489	0.551
	+ both	**0.437**	0.507	0.561	0.590	0.614	0.382	0.457
F	no boundaries	0.329	0.343	0.440	**0.459**	0.429	**0.277**	**0.339**
	95%-Conf.	*±0.020*	*±0.022*	*±0.025*	*±0.023*	*±0.039*	*±0.025*	*±0.028*
	+ initial	**0.337**	**0.345**	0.437	0.458	**0.447**	0.176	0.252
	+ end	0.330	0.341	**0.441**	0.456	0.425	0.226	0.291
	+ both	0.334	0.340	0.437	0.448	0.437	0.130	0.205
A+F	no boundaries	0.427	**0.498**	0.563	**0.572**	0.560	**0.448**	**0.514**
	95%-Conf.	*±0.018*	*±0.019*	*±0.022*	*±0.020*	*±0.033*	*±0.021*	*±0.023*
	+ initial	0.431	0.494	0.558	0.567	**0.577**	0.333	0.417
	+ end	0.429	0.497	**0.565**	0.569	0.558	0.403	0.472
	+ both	**0.432**	0.493	0.561	0.560	0.570	0.290	0.373

Table B.16: Correlation of different block move distances with human evaluation on 20 word-corpora. Pearson's r on sentence level.

Human score	Automatic evaluation measure	TIDES 2002 CE	TIDES 2003 CE	TIDES 2003 AE	TIDES 2004 CE	TIDES 2004 AE	BTEC 2004 CE	BTEC 2004 JE
A	BLEU	0.495	0.471	0.606	0.601	0.646	0.630	0.693
	95%-Conf.	*±0.037*	*±0.042*	*±0.041*	*±0.047*	*±0.055*	*±0.016*	*±0.016*
	WER	0.398	0.391	0.537	0.549	0.662	0.686	0.740
	CDER	**0.519**	**0.497**	**0.615**	**0.630**	**0.692**	0.696	0.746
	LJWER	0.405	0.416	0.555	0.576	0.664	0.709	0.755
	INVWER	0.416	0.428	0.570	0.593	0.661	**0.712**	**0.759**
F	BLEU	0.408	**0.376**	**0.486**	0.525	0.533	0.477	0.555
	95%-Conf.	*±0.041*	*±0.047*	*±0.050*	*±0.054*	*±0.069*	*±0.021*	*±0.021*
	WER	0.342	0.324	0.435	0.521	0.514	**0.548**	**0.616**
	CDER	**0.445**	0.346	0.467	**0.548**	**0.536**	0.492	0.538
	LJWER	0.337	0.333	0.451	0.527	0.517	0.531	0.584
	INVWER	0.330	0.325	0.460	0.532	0.504	0.531	0.590
A+F	BLEU	0.495	0.477	**0.612**	0.599	0.636	0.622	0.698
	95%-Conf.	*±0.037*	*±0.042*	*±0.040*	*±0.047*	*±0.057*	*±0.016*	*±0.016*
	WER	0.402	0.400	0.546	0.567	0.637	0.693	**0.757**
	CDER	**0.527**	**0.482**	0.609	**0.628**	**0.666**	0.669	0.719
	LJWER	0.405	0.422	0.564	0.586	0.639	0.697	0.749
	INVWER	0.409	0.426	0.578	0.599	0.631	**0.699**	0.754

Table B.17: Effect of different application directions on the correlation between CDER and human evaluation. Pearson's r on sentence level.

Human score	Direction	TIDES 2002 CE	TIDES 2003 CE	TIDES 2003 AE	TIDES 2004 CE	TIDES 2004 AE	BTEC 2004 CE	BTEC 2004 JE
A	CD for candidate	**0.411**	**0.449**	**0.535**	**0.615**	**0.630**	0.703	0.750
	95%-Conf.	*±0.018*	*±0.019*	*±0.022*	*±0.018*	*±0.028*	*±0.013*	*±0.013*
	CD for reference	0.150	0.145	0.394	0.197	0.382	0.447	0.472
	sum of both	0.295	0.321	0.501	0.460	0.563	0.679	0.711
	maximum of both	0.331	0.391	0.509	0.580	0.612	**0.715**	**0.762**
F	CD for candidate	**0.363**	**0.357**	**0.440**	**0.557**	**0.525**	0.511	0.550
	95%-Conf.	*±0.019*	*±0.021*	*±0.025*	*±0.020*	*±0.033*	*±0.019*	*±0.021*
	CD for reference	0.142	0.179	0.369	0.228	0.351	0.540	0.551
	sum of both	0.266	0.298	0.434	0.449	0.482	**0.614**	**0.633**
	maximum of both	0.281	0.321	0.427	0.534	0.499	0.539	0.583
A+F	CD for candidate	**0.424**	**0.458**	**0.544**	**0.625**	**0.623**	0.680	0.724
	95%-Conf.	*±0.018*	*±0.019*	*±0.022*	*±0.018*	*±0.028*	*±0.014*	*±0.014*
	CD for reference	0.157	0.171	0.422	0.222	0.393	0.548	0.565
	sum of both	0.305	0.343	0.521	0.482	0.563	**0.722**	0.747
	maximum of both	0.337	0.402	0.523	0.594	0.599	0.702	**0.749**

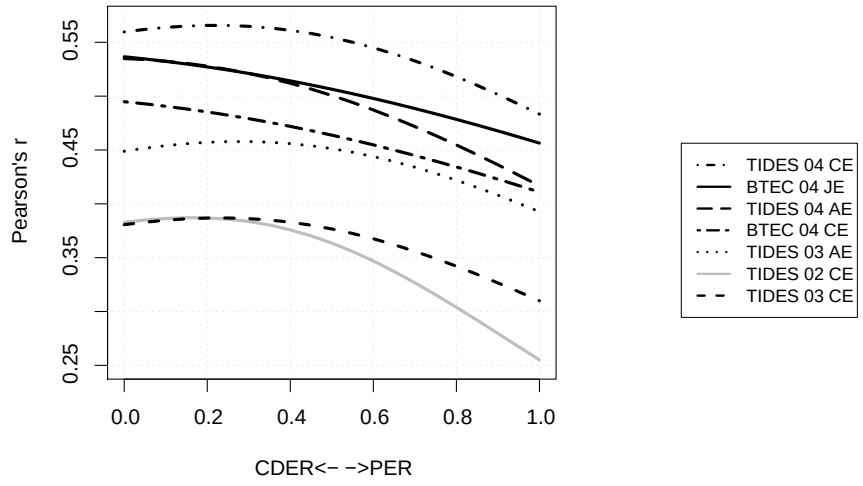


Figure B.1: Effects of the weights on the correlation between fluency and a weighted linear combination of CDER and PER. Pearson's r on sentence level.

Table B.18: Effect of "Boundaries" on the correlation between CDER and human evaluation. Pearson's r on sentence level.

Human score	sentence boundaries	TIDES 2002 CE	TIDES 2003 CE	TIDES 2003 AE	TIDES 2004 CE	TIDES 2004 AE	BTEC 2004 CE	BTEC 2004 JE
A	no boundaries	0.414	0.449	0.546	0.614	0.624	0.708	**0.761**
	95%-Conf.	*±0.018*	*±0.019*	*±0.021*	*±0.018*	*±0.028*	*±0.013*	*±0.013*
	+ initial	**0.417**	**0.455**	**0.546**	**0.617**	0.630	**0.714**	0.758
	+ end	0.411	0.444	0.536	0.613	0.626	0.699	0.753
	+ both	0.411	0.449	0.535	0.615	**0.630**	0.703	0.750
F	no boundaries	0.346	0.349	0.444	0.552	0.507	**0.549**	**0.601**
	95%-Conf.	*±0.019*	*±0.021*	*±0.025*	*±0.020*	*±0.034*	*±0.018*	*±0.019*
	+ initial	0.359	**0.358**	**0.447**	0.556	0.521	0.538	0.576
	+ end	0.352	0.350	0.439	0.554	0.513	0.524	0.578
	+ both	**0.363**	0.357	0.440	**0.557**	**0.525**	0.511	0.550
A+F	no boundaries	0.419	0.455	0.554	0.623	0.611	**0.703**	**0.758**
	95%-Conf.	*±0.018*	*±0.019*	*±0.021*	*±0.018*	*±0.029*	*±0.013*	*±0.013*
	+ initial	**0.427**	**0.462**	**0.555**	**0.626**	0.621	0.701	0.743
	+ end	0.420	0.452	0.545	0.623	0.615	0.684	0.741
	+ both	0.424	0.458	0.544	0.625	**0.623**	0.680	0.724

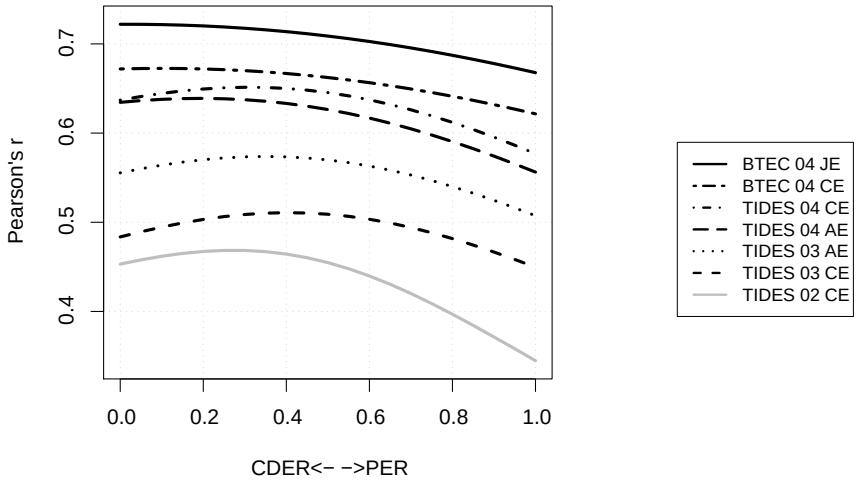


Figure B.2: Effects of the weights on the correlation adequacy plus fluency and a weighted linear combination of CDER and PER. Pearson's r on sentence level.

Table B.19: Effect of different miscoverage penalty functions on the correlation between CDER and human evaluation. Pearson's r on sentence level.

Human score	CDER miscoverage penalty	TIDES 2002 CE	TIDES 2003 CE	TIDES 2003 AE	TIDES 2004 CE	TIDES 2004 AE	BTEC 2004 CE	BTEC 2004 JE
A	CDER	**0.411**	**0.449**	**0.535**	**0.615**	**0.630**	0.703	0.750
	95%-Conf.	*±0.018*	*±0.019*	*±0.022*	*±0.018*	*±0.028*	*±0.013*	*±0.013*
	+ path miscoverage	0.271	0.293	0.456	0.448	0.533	0.673	0.720
	+ length difference	0.296	0.365	0.459	0.567	0.582	**0.711**	**0.758**
	+ $\frac{1}{2}$ both	0.293	0.334	0.470	0.517	0.567	0.703	0.747
F	CDER	**0.363**	**0.357**	**0.440**	**0.557**	**0.525**	0.511	0.550
	95%-Conf.	*±0.019*	*±0.021*	*±0.025*	*±0.020*	*±0.033*	*±0.019*	*±0.021*
	+ path miscoverage	0.237	0.262	0.382	0.428	0.448	0.539	0.575
	+ length difference	0.251	0.305	0.388	0.522	0.467	0.537	0.581
	+ $\frac{1}{2}$ both	0.253	0.289	0.395	0.486	0.465	**0.548**	**0.583**
A+F	CDER	**0.424**	**0.458**	**0.544**	**0.625**	**0.623**	0.680	0.724
	95%-Conf.	*±0.018*	*±0.019*	*±0.022*	*±0.018*	*±0.028*	*±0.014*	*±0.014*
	+ path miscoverage	0.277	0.309	0.469	0.466	0.528	0.677	0.721
	+ length difference	0.300	0.377	0.473	0.581	0.567	0.698	**0.746**
	+ $\frac{1}{2}$ both	0.299	0.349	0.483	0.534	0.557	**0.700**	0.741

Table B.20: Effect of word-dependent substitution costs on the correlation between WER and human evaluation. Pearson's r on sentence level.

Human score	c_{SUB} depending on	TIDES 2002 CE	TIDES 2003 CE	TIDES 2003 AE	TIDES 2004 CE	TIDES 2004 AE	BTEC 2004 CE	BTEC 2004 JE
A	WER	0.320	0.349	0.505	0.540	0.597	0.691	0.744
	95%-Conf.	*±0.020*	*±0.021*	*±0.023*	*±0.020*	*±0.029*	*±0.014*	*±0.014*
	+ prefix	**0.349**	**0.373**	**0.516**	0.554	0.613	**0.692**	**0.752**
	+ Levenshtein	0.346	0.371	0.514	**0.564**	**0.616**	0.682	0.746
F	WER	0.277	0.301	0.423	0.511	0.496	**0.565**	**0.624**
	95%-Conf.	*±0.020*	*±0.022*	*±0.025*	*±0.021*	*±0.035*	*±0.018*	*±0.019*
	+ prefix	0.299	**0.322**	0.432	0.517	0.509	0.552	0.612
	+ Levenshtein	**0.299**	0.321	**0.437**	**0.525**	**0.518**	0.535	0.600
A+F	WER	0.328	0.365	0.518	0.559	0.589	**0.702**	**0.761**
	95%-Conf.	*±0.020*	*±0.021*	*±0.022*	*±0.020*	*±0.030*	*±0.013*	*±0.013*
	+ prefix	**0.356**	**0.389**	0.530	0.571	0.605	0.695	0.759
	+ Levenshtein	0.354	0.388	**0.531**	**0.580**	**0.611**	0.681	0.750

Table B.21: Effect of word-dependent substitution costs on the correlation between PER and human evaluation. Pearson's r on sentence level.

Human score	c_{SUB} depending on	TIDES 2002 CE	TIDES 2003 CE	TIDES 2003 AE	TIDES 2004 CE	TIDES 2004 AE	BTEC 2004 CE	BTEC 2004 JE
A	PER	0.329	0.428	0.495	0.579	0.600	**0.708**	**0.744**
	95%-Conf.	*±0.020*	*±0.020*	*±0.023*	*±0.019*	*±0.029*	*±0.013*	*±0.014*
	+ prefix	**0.359**	**0.464**	**0.508**	**0.589**	**0.606**	0.696	0.738
	+ Levenshtein	0.334	0.452	0.488	0.567	0.593	0.663	0.711
F	PER	0.245	0.298	0.389	**0.493**	**0.424**	**0.456**	**0.504**
	95%-Conf.	*±0.021*	*±0.022*	*±0.026*	*±0.022*	*±0.038*	*±0.021*	*±0.023*
	+ prefix	**0.255**	**0.310**	**0.393**	0.483	0.417	0.411	0.456
	+ Levenshtein	0.234	0.293	0.380	0.460	0.405	0.364	0.422
A+F	PER	0.321	0.419	0.496	0.575	**0.556**	**0.653**	**0.697**
	95%-Conf.	*±0.020*	*±0.020*	*±0.023*	*±0.019*	*±0.032*	*±0.015*	*±0.016*
	+ prefix	**0.345**	**0.450**	**0.507**	**0.577**	0.556	0.622	0.668
	+ Levenshtein	0.320	0.435	0.489	0.553	0.542	0.578	0.634

Table B.22: Effect of word-dependent substitution costs on the correlation between CDER and human evaluation. Pearson's r on sentence level.

Human score	c_{SUB} depending on	TIDES 2002 CE	TIDES 2003 CE	TIDES 2003 AE	TIDES 2004 CE	TIDES 2004 AE	BTEC 2004 CE	BTEC 2004 JE
A	CDER	0.411	0.449	0.535	0.615	0.630	0.703	0.750
	95%-Conf.	*±0.018*	*±0.019*	*±0.022*	*±0.018*	*±0.028*	*±0.013*	*±0.013*
	+ prefix	0.442	0.472	0.545	**0.631**	0.642	**0.705**	**0.757**
	+ Levenshtein	**0.442**	**0.472**	**0.547**	0.630	**0.643**	0.696	0.749
F	CDER	0.363	0.357	0.440	0.557	0.525	**0.511**	**0.550**
	95%-Conf.	*±0.019*	*±0.021*	*±0.025*	*±0.020*	*±0.033*	*±0.019*	*±0.021*
	+ prefix	0.383	0.381	0.449	0.560	0.535	0.495	0.537
	+ Levenshtein	**0.391**	**0.381**	**0.455**	**0.563**	**0.539**	0.479	0.524
A+F	CDER	0.424	0.458	0.544	0.625	0.623	**0.680**	**0.724**
	95%-Conf.	*±0.018*	*±0.019*	*±0.022*	*±0.018*	*±0.028*	*±0.014*	*±0.014*
	+ prefix	0.453	0.484	0.555	0.637	0.634	0.672	0.722
	+ Levenshtein	**0.457**	**0.484**	**0.559**	**0.638**	**0.637**	0.658	0.710

Appendix C

Notation and proofs

This appendix gives an overview of the notation within this thesis. Moreover, it contains additional proofs and derivations not found within the references.

Table C.1: Notation of sentences.

Symbol	*Description*
E_k	Candidate sentence in test set. $k = 1, \dots, K$
$\widetilde{E}_{r,k}$	Reference sentence for candidate sentence E_k. For each E_k, there are $r = 1, \dots, R_k$ reference sentences.
I_k	Length of candidate sentence E_k.
$L_{k,r}$	Length of reference sentence $\widetilde{E}_{r,k}$.
L^*_k	Reference length for candidate sentence E_k.
I_{tot}	Total candidate length over the corpus. $I_{tot} := \sum_k I_k$ (C.1)
L^*_{tot}	Total reference length over the corpus. $L^*_{tot} := \sum_k L^*_k$ (C.2)

Table C.2: Notation of words and m-grams.

Symbol	*Description*
ε	The empty word
e	word
e_i	word at position i in candidate sentence
$\tilde{e}_l$	word at position l in reference sentence
$e_i^{i'}$	substring from position i to position i' in candidate sentence
$\tilde{e}_l^{l'}$	substring from position l to position l' in reference sentence
e^m	m-gram

Table C.3: Notation of count vectors.

Symbol	*Description*
$n_{e^m,k}$	Count of m-gram e^m in candidate sentence E_k
$\widetilde{n}_{e^m,r,k}$	Count of m-gram e^m in reference sentence $\widetilde{E}_{r,k}$
$n^{\cap}_{e^m,k}$	Co-occurrence count of m-gram e^m in E_k and $\widetilde{E}_{r,k}$. See Equation (3.2).
$n^{\cap}_m$	Total co-occurrence count of all m-gram. See Equation (3.3).
N_m	Total m-gram count in candidate corpus.
	$N_m := \sum_k \sum_{e^m \subseteq E_k} n_{e^m,k}$ (C.3)
$\widetilde{N}_m$	Total m-gram count in candidate corpus.
	$\widetilde{N}_m := \sum_k \sum_r \sum_{e^m \subseteq \widetilde{E}_{r,k}} \widetilde{n}_{e^m,r,k}$. (C.4)

Table C.4: Notation of operations and functions.

Symbol	*Description*
op	Edit operation
c_{op}	Cost of an edit operation or grammar rule
$c_{op}(e, \tilde{e})$	Word dependent cost of an edit operation
lp	Length penalty
$\delta(e, \tilde{e})$	Kronecker-delta: $\delta(e, \tilde{e}) := \begin{cases} 0 & \text{if } e = \tilde{e} \\ 1 & \text{otherwise} \end{cases}$ (C.5)

Table C.5: Notation of random variables, covariance, etc.

Symbol	*Description*
X	Random variable
x_i	Sample #i of random variable X
μ_X	Mean of X. $\mu_X := \frac{1}{N} \sum_i x_i$ (C.6)
σ_{XX}	Variance of X. $\sigma_{XX} := \frac{1}{N} \sum_i (x_i - \mu_X)^2$ (C.7)
σ_X	Standard deviation of X. $\sigma_X := \sqrt{\sigma_{XX}}$ (C.8)
σ_{XY}	Covariance of X and Y. $\sigma_{XY} := \frac{1}{N} \sum_i (x_i - \mu_X)(y_i - \mu_Y)$ (C.9)

Table C.6: Notation of bracketing transduction grammars.

Symbol	*Description*
A	Nonterminal symbol
S	Start symbol
$[AA]$	Straight concatenation
$\langle AA \rangle$	Inverted concatenation
α, β	Parse subtrees
$c(\alpha)$	Total parse costs of α

C.1 Proof: r and linear regression

According to [Casella & Berger 90] (12.2.4a),

$$\mathrm{E}\left[\,\|Y-\hat{Y}\|^2\,\right] = \sigma_{YY} - 2\beta\sigma_{XY} + \beta^2\sigma_{XX}. \tag{C.10}$$

With $\beta = \frac{\sigma_{XY}}{\sigma_{XX}}$ [Casella & Berger 90] (12.2.5),

$$\mathrm{E}\left[\,\|Y-\hat{Y}\|^2\,\right] = \sigma_{YY} - 2\frac{{\sigma_{XY}}^2}{\sigma_{XX}} + \frac{{\sigma_{XY}}^2}{\sigma_{XX}} \tag{C.11}$$

$$= \sigma_{YY}\left(1 - \frac{{\sigma_{XY}}^2}{\sigma_{XX}\sigma_{YY}}\right) \qquad \square \tag{C.12}$$

By symmetry, the same holds for a regression of X given Y.

C.2 Proof: r and least orthogonal squares regression

According to [Casella & Berger 90] (12.3.10a),

$$\mathrm{E}\left[\,\|(X,Y)-(\hat{X},\hat{Y})\|^2\,\right] = \frac{1}{1+\beta^2}\left(\sigma_{YY} - 2\beta\sigma_{XY} + \beta^2\sigma XX\right) \tag{C.13}$$

where β is the slope of the LOS regression line. For $\sigma_{XX} = \sigma_{YY}$, the slope becomes $\beta = \pm 1$. Assume $\beta = 1$, i.e. $r > 0$. Then, (C.13) can be simplified to

$$\mathrm{E}\left[\,\|(X,Y)-(\hat{X},\hat{Y})\|^2\,\right] = \frac{1}{1+\beta^2}\left(\sigma_{YY} - 2\beta\sigma_{XY} + \beta^2\sigma XX\right) \tag{C.14}$$

$$= \frac{1}{2}\left(2\sigma^2 - 2\sigma_{XY}\right) \tag{C.15}$$

$$= \sigma^2\left(1 - \frac{\sigma_{XY}}{\sigma^2}\right) \qquad \square \tag{C.16}$$

Analogously for $\beta = -1$, i.e. $r < 0$.

C.3 Proof: r and linear combination of probability variables

$$\begin{aligned}
\sigma_{X_w X_w} &= \frac{1}{N}\sum \left(wx_1 + (1-w)x_2 - (w\bar{x}_1 + (1-w)\bar{x}_2)\right)^2 \\
&= \frac{1}{N}\sum \left(w\left(x_1 - \bar{x}_1\right) + (1-w)\left(x_2 - \bar{x}_2\right)\right)^2 \\
&= w^2 \frac{1}{N}\sum \left(x_1 - \bar{x}_1\right)^2 \\
&\quad + 2w(1-w)\frac{1}{N}\sum \left(x_1 - \bar{x}_1\right)\left(x_2 - \bar{x}_2\right) \\
&\quad + (1-w)^2 \frac{1}{N}\sum \left(x_2 - \bar{x}_2\right)^2 \\
&= w^2\sigma_{X_1 X_1} + 2w(1-w)\sigma_{X_1 X_2} + (1-w)^2\sigma_{X_2 X_2} \qquad \text{(C.17)}
\end{aligned}$$

$$\begin{aligned}
\sigma_{X_w Y} &= \frac{1}{N}\sum \left(wx_1 + (1-w)x_2 - (w\bar{x}_1 + (1-w)\bar{x}_2)\right)\left(y - \bar{y}\right) && \text{(C.18)} \\
&= \frac{1}{N}\sum \left(wx_1 - w\bar{x}_1 + (1-w)x_2 - (1-w)\bar{x}_2\right)\left(y - \bar{y}\right) && \text{(C.19)} \\
&= w\frac{1}{N}\sum \left(x_1 - \bar{x}_1\right)\left(y - \bar{y}\right) + (1-w)\frac{1}{N}\sum \left(x_2 - \bar{x}_2\right)\left(y - \bar{y}\right) && \text{(C.20)} \\
&= w\sigma_{X_1 Y} + (1-w)\sigma_{X_2 Y} && \text{(C.21)} \\
& && \text{(C.22)}
\end{aligned}$$

Bibliography

[Akiba & Federico$^+$ 04] Y. Akiba, M. Federico, N. Kando, H. Nakaiwa, M. Paul, J. Tsujii: Overview of the IWSLT04 Evaluation Campaign. *Proc. IWSLT*, pp. 1–12, Kyoto, Japan, September 2004.

[Akiba & Imamura$^+$ 01] Y. Akiba, K. Imamura, E. Sumita: Using Multiple Edit Distances to Automatically Rank Machine Translation Output. *Proc. MT-Summit VIII*, pp. 15–20, September 2001.

[Arnold & Balkan$^+$ 94] D. Arnold, L. Balkan, S. Meijer, L.L. Humphreys, L. Sadler: *Machine Translation: An Introductory Guide*, chapter 9, pp. 165–181. Blackwells-NCC, London, 1994.

[Beazley & Fulton$^+$ 02] D. Beazley, W. Fulton, M. Köppe, L. Johnson, R. Palmer: Simplified Wrapper and Interface Generator. `http://www.swig.org/`, 2002.

[Bisani & Ney 04] M. Bisani, H. Ney: Bootstrap Estimates for Confidence Intervals in ASR Performance Evaluation. *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 409–412, Montreal, Canada, May 2004.

[Blatz & Fitzgerald$^+$ 03] J. Blatz, E. Fitzgerald, G. Foster, S. Gandrabur, C. Goutte, A. Kulesza, A. Sanchis, N. Ueffing: Confidence Estimation for Machine Translation. Final report, JHU/CLSP Summer Workshop, 2003. `http://www.clsp.jhu.edu/ws2003/groups/ estimate/`.

[BTEC 04] Basic Travel Expression Corpus, 2004. `http://cstar.atr.jp/cstar-corpus/`.

[Casella & Berger 90] G. Casella, R.L. Berger: *Statistical Inference*, chapter 4.5, pp. 160–168. Duxbury Press, 1990.

[Ciery & Huang$^+$ 02] C. Ciery, S. Huang, M. Babma, K. Walker, D. Graff: Multiple Human Translations and Other Resources for MT Development and Evaluation. *TIDES Machine Translation Workshop*, Marina del Rey, CA, January 2002.

[Cormen & Leiserson$^+$ 90] T.H. Cormen, C.E. Leiserson, R.L. Rivest: *Introduction to Algorithms*, chapter 16, pp. 301–328. MIT Press/McGraw-Hill, 1990.

[Doddington 02] G. Doddington: Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. *Proc. ARPA Workshop on Human Language Technology*, 2002.

[Doddington 03] G. Doddington: NIST MT Evaluation Workshop. Personal communication, July 2003.

[EAG 96] *EAGLES Evaluation Group. Final Report*. Copenhagen, Denmark, October 1996.

[Efron & Tibshirani 93] B. Efron, R.J. Tibshirani: *An Introduction to the Bootstrap*. Chapman & Hall, New York and London, 1993.

[Held & Karp 62] M. Held, R.M. Karp: A dynamic programming approach to sequencing problems. *Journal of the Society of Industrial and Applied Mathematics*, Vol. 10, pp. 196–210, 1962.

[Kendall 70] M.G. Kendall: *Rank Correlation Methods*. Charles Griffin & Co Ltd, London, 1970.

[Knuth 93] D.E. Knuth: *The Stanford GraphBase: a platform for combinatorial computing*. ACM Press, New York, NY, USA, 1993.

[LDC 05] LDC: Linguistic Data Annotation Specification: Assessment of Fluency and Adequacy in Chinese-English Translations, Revision 1.5. `http://www.ldc.upenn.edu/Projects/TIDES/Translation/TransAssess04.pdf`, 2005.

[Leusch & Nießen$^+$ 02] G. Leusch, S. Nießen, R. Zens, H. Ney: The `EvalTrans` Machine Translation Evaluation Software, 2002. `http://www-i6.informatik.rwth-aachen.de/web/Software/EvalTrans/`.

[Leusch & Ueffing$^+$ 03] G. Leusch, N. Ueffing, H. Ney: A Novel String-to-String Distance Measure with Applications to Machine Translation Evaluation. *Proc. MT Summit IX*, pp. 240–247, New Orleans, LA, September 2003.

[Leusch & Ueffing$^+$ 05] G. Leusch, N. Ueffing, D. Vilar, H. Ney: Preprocessing and Normalization for Automatic Evaluation of Machine Translation. *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pp. 17–24, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics.

[Levenshtein 66] V.I. Levenshtein: Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, Vol. 10, No. 8, pp. 707–710, Feb. 1966.

[Lin & Och 04a] C.Y. Lin, F.J. Och: Automatic Evaluation of Machine Translation Quality Using Longest Common Subsequence and Skip-Bigram Statistics. *Proc. ACL 2004*, pp. 605–612, 2004.

[Lin & Och 04b] C.Y. Lin, F.J. Och: ORANGE: a method for evaluation automatic evaluation metrics for machine translation. *Proc. COLING 2004*, pp. 501–507, Geneva, Switzerland, August 2004.

[Lopresti & Tomkins 97] D. Lopresti, A. Tomkins: Block edit models for approximate string matching. *Theoretical Computer Science*, Vol. 181, No. 1, pp. 159–179, July 1997.

[Michie 68] D. Michie: Memo Functions and Machine Learning. *Nature*, Vol. 281, No. 1, pp. 19–22, April 1968.

[Nießen & Och$^+$ 00] S. Nießen, F.J. Och, G. Leusch, H. Ney: An evaluation tool for machine translation: Fast evaluation for MT research. *Proc. of the Second Int. Conf. on Language Resources and Evaluation (LREC)*, pp. 39–45, Athens, Greece, May 2000.

[NIST 02] NIST: MT Evaluation Chinese–English. `http://nist.gov/speech/tests/mt/`, 2002.

[Ousterhout & TclCoreTeam 04] J. Ousterhout, TclCoreTeam: The Tcl/Tk language, Version 8.4.7, 2004. `http://www.tcl.tk`.

[Papineni 02] K.A. Papineni: The NIST `mteval` scoring software, 2002. `http://www.itl.nist.gov/iad/894.01/tests/mt/resources/scoring.htm`.

[Papineni & Roukos$^+$ 01] K.A. Papineni, S. Roukos, T. Ward, W.J. Zhu: Bleu: a Method for Automatic Evaluation of Machine Translation. Technical Report RC22176 (W0109-022), IBM Research Division, Thomas J. Watson Research Center, 10 pages, September 2001.

[Papineni & Roukos$^+$ 02] K. Papineni, S. Roukos, T. Ward, J. Henderson, F. Reeder: Corpus-based comprehensive and diagnostic MT evaluation: Initial Arabic, Chinese, French, and Spanish results. *Proc. of the Human Language Technology Conf.*, pp. 124–127, San Diego, CA, March 2002.

[Paul & Nakaiwa$^+$ 04] M. Paul, H. Nakaiwa, M. Federico: Towards Innovative Evaluation Methodologies for Speech Translation. *Working Notes of the NTCIR-4 Meeting*, Vol. 2, pp. 17–21, 2004.

[Pierce & Carroll$^+$ 66] J. Pierce, J. Carroll, E. Hamp, D. Hays, C. Hockett: *Languages and machines: computers in translation and linguistics (ALPAC)*. Number 1416. Washington, D.C.: National Academy of Sciences, National Research Council, 1966.

[Popescu-Belis & Manzi$^+$ 01] A. Popescu-Belis, S. Manzi, M. King: Towards a Two-stage Taxonomy for Machine Translation Evaluation. *Workshop on Machine Translation Evaluation at MT Summit VIII*, pp. 1–8, Santiago de Compostela, Spain, September 2001.

[Przybocki 03] M. Przybocki: NIST 2003 Machine Translation Evaluation. *Machine Translation Evaluation Workshop*, Gaithersburg, MD, July 2003.

[Przybocki 04] M. Przybocki: NIST Machine Translation 2004 Evaluation: Summary of Results. *Machine Translation Evaluation Workshop*, Alexandria, Virginia, June 2004.

[Rubner & Tomasi$^+$ 98] Y. Rubner, C. Tomasi, L.J. Guibas: A Metric for Distributions with Applications to Image Databases. *ICCV '98: Proceedings of the Sixth International Conference on Computer Vision*, 59, Washington, DC, USA, 1998. IEEE Computer Society.

[Siegel & Castellan 88] S. Siegel, N. Castellan: *Nonparametric statistics for the behavioral sciences*. McGraw-Hill, New York, NY, USA, 2nd edition, 1988.

[Tillmann & Ney 00] C. Tillmann, H. Ney: Word re-ordering and DP-based search in statistical machine translation. *COLING '00: The 18th Int. Conf. on Computational Linguistics*, pp. 850–856, Saarbrücken, Germany, July 2000.

[Tillmann & Vogel$^+$ 97] C. Tillmann, S. Vogel, H. Ney, A. Zubiaga, H. Sawaf: Accelerated DP Based Search for Statistical Translation. *European Conf. on Speech Communication and Technology*, pp. 2667–2670, Rhodes, Greece, September 1997.

[Turian & Shen$^+$ 03] J.P. Turian, L. Shen, I.D. Melamed: Evaluation of machine translation and its evaluation. *Proc. MT Summit IX*, pp. 23–28, New Orleans, LA, September 2003.

[Vogel & Nießen$^+$ 00] S. Vogel, S. Nießen, H. Ney: Automatic Extrapolation of Human Assessment of Translation Quality. *2nd Int. Conf. on Language Resources and Evaluation (LREC 2000): Proc. of the Workshop on Evaluation of Machine Translation*, pp. 35–39, Athens, Greece, May 2000.

[Wu 95] D. Wu: An Algorithm for Simultaneously Bracketing Parallel Texts by Aligning Words. *Proc. of the 33rd Annual Conf. of the Association for Computational Linguistics*, pp. 244–251, 1995.

[Younger 67] D. Younger: Recognition and parsing of context-free languages in time n^3. *Information and Control*, Vol. 10, No. 2, pp. 189–208, 1967.

Index