BLEUSP, INVWER, CDER: Three improved MT evaluation measures

Gregor Leusch and **Hermann Ney**
RWTH Aachen University, Germany
{leusch,ney}@cs.rwth-aachen.de

Abstract

We present three modifications of well-established automatic machine translation evaluation measures, to improve correlation between those measures and human evaluation. Following Lin & Och, we present an improved version of the BLEU score, which uses a smoothed geometric mean for combining different n-gram precisions. We use segment boundary markers to increase the weight of words near the segment boundaries in the BLEU score. Our second MT evaluation measure is a variant of the WER which allows for block movements, but does not demand complete and disjoint coverage of the source sentence. As this might be problematic if MT systems are tuned on this score, we later investigate a linear combination of this measure with PER. Finally, we describe an edit distance similar to TER, which also allows for block reordering. Our measure uses a full search, but with the constraint that block operations must be bracketed. We describe this measure using a Bracketing Transduction Grammar, and sketch a polynomial-time algorithm for its calculation. We also modify the WER-like measures such that they use word-dependent substitution costs instead of fixed ones to model the similarity between words. Experimental comparison of these measures show that our new measures correlate significantly better with human judgment than the original measures.

1 Introduction

For a couple of reasons, automatic evaluation of Machine Translation (MT) systems is a difficult task, mostly because it is difficult to define when a translation is good, and when it is bad. Or which of two given translations is better, and which one is worse. The main reason for this are ambiguities in natural languages: Usually, there is more than one correct translation for a source sentences; there are ambiguities in the choice of synonyms as well as in the order of the words. Because of the difficulty of this task, a multitude of automatic MT evaluation measures have been defined over the last couple of years. Some of these measures have become well-established, for example BLEU or TER, others are only of medium or small significance. We expect that in the context of NIST's Metrics MATR evaluation, more measures will be added to the pool of evaluation measures.

In a few previous papers, the proposed measures seemed to be more of theoretical interest than of practical use: While they certainly emphasis important linguistic effects, it is not investigated systematically in how far these effects play a role in the difference in quality in different MT systems. Some other proposed evaluation measures seemed to focus on specific properties and features of the previously generated translations they are trained or optimized on, which can (but does not need to) lead to evaluation measures which are basically classifiers dividing previously good from previously poor systems, or "easy" from "difficult" source sentences. If measures with this property are used to tune a typical statistical MT system, it can sometimes be observed that the MT system learns to "play" against this, and might even learn to produce translations which show the "good" features without actually being good translations. For example, Rosti et al. (2007) report such an effect. This is not to say that all new measures share these problems, nor that there is no need for MT evaluation measures which go beyond lexical comparison – quite the opposite. But these issues were the motivation for us to start from established evaluation measures, with known properties especially with regard to tuning, and alter them at a few selected points to improve their correlation with human judgment.

This paper is organized as follows: In Section 2, we describe some modifications to the BLEU score, following Lin and Och (2004), and Leusch et al. (2005). We present a simple variant of WER in Section 3, called CDER, which allows for block transposition similar to TER, following Leusch et al. (2006). This measure can be efficiently calculated exactly, without having to resort to shift heuristics or greedy search as in TER. The tradeoff is that this measure by itself measures basically recall, not precision. To overcome this bias, we will later propose a linear combination of this measure and PER in Section 6. Before this, in Section 4, we describe another variant of TER which can be exactly calculated in polynomial time, this time by restricting possible shifts to ITG constrains. This method follows Leusch et al. (2003). We call this measure INVWER. In Section 5, we introduce two simple methods following Leusch et al. (2006) to improve edit-operation–based measures like PER, WER, TER, and CDER/INVWER by taking into account the lexical difference of words in a substitution operation. After an experimental evaluation of our three proposed evaluation measures in Section 7, we conclude this paper in Section 8.

2 BLEUSP

BLEU (Papineni et al., 2001) is a precision measure based on n-gram count vectors. The precision is modified such that multiple references are combined into a single n-gram count vector. Multiple occurrences of an n-gram in the candidate sentence are counted as correct only up to the maximum occurrence count within the reference sentences. Typically, unigrams, bigrams, trigrams, and four-grams are used for BLEU; their four precisions are combined using the geometric mean.

To avoid a bias towards short candidate sentences consisting of "safe guesses" only, sentences shorter than the reference length will be penalized with a brevity penalty.

In the original BLEU definition there is no smoothing for the geometric mean. This has the disadvantage that the whole score becomes zero already if the four-gram precision is zero, which especially happens often with short or difficult translations. As a result, sentence-level scores are often quite noisy, and not usable for evaluation. To allow for sentence-wise evaluation, Lin and Och (2004) define the BLEUS measure, which is basically BLEU where all bi-, tri-, and four-gram counts are initialized with 1 instead of 0. We have adopted this technique for this study, as experiments showed a clear improvement over BLEU in terms of correlation with human judgment on segment and document level; the effect on system level scores is negligible due to the already high n-gram counts here.

In our experiments, BLEU and BLEUS lack in another minor point: The position of a word within a sentence can be quite significant for the correctness of the sentence. WER, TER, and CDER/INVWER (Sections 3 and 4) explicitly take into account the ordering of the words in a sentence. This is not the case with BLEU, although the order of inner words is regarded implicitly by n-gram overlap. To model the position of words at the initial or the end of a sentence, we enclose the sentence with artificial sentence boundary tokens.

For example, the sentence `A B C` is considered to consist of

- the unigrams `[A]`, `[B]`, and `[C]`,
- the bigrams `[<s> A]`, `[A B]`, `[B C]`, and `[C </s>]`,
- the trigrams `[<s> <s> A]`, `[<s> A B]`, `[A B C]`, `[B C </s>]`, and `[C </s> </s>]`,
- etc.

In the measure we denote as BLEUSP, all n-grams are counted like this for all candidate and reference segments, and for these counts, the BLEUS score is calculated.

3 CDER

As translations of sentences are often ambiguous in the order of phrases, reorderings of whole blocks of

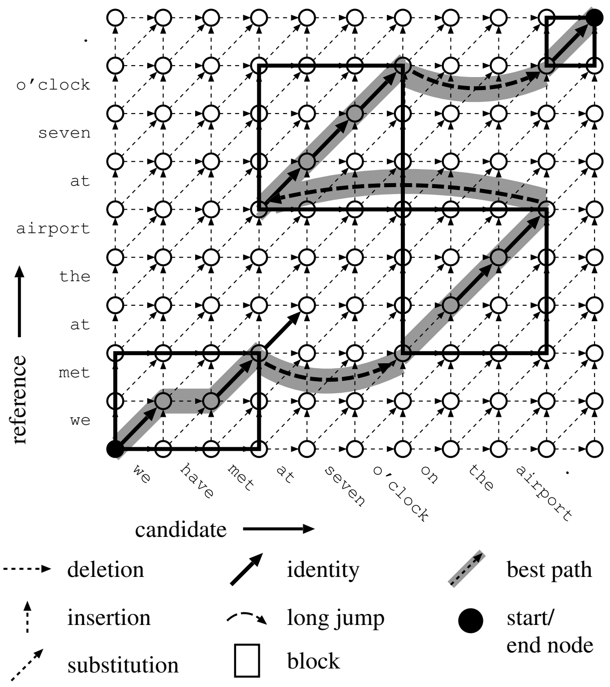

Figure 1: Example of a long jump alignment grid. All possible deletion, insertion, identity and substitution operations are depicted. Only long jump edges from the best path are drawn.

words should not be penalized too hard by an MT evaluation measure. WER, which is based on the classical Levenshtein distance (Levenshtein, 1966), penalizes block reorderings rather hard – each word that has been shifted usually needs to be deleted in its old position, and inserted in its new position. One approach here is to extend the Levenshtein distance by an additional operation, namely *block movement* (or *shift*, as it is called in TER (Snover et al., 2005)). Note that the number of blocks in a sentence is equal to the number of gaps among the blocks plus one. Thus, the block movements can equivalently be expressed as *long jump* operations that jump over the gaps between two blocks. The costs of a long jump are considered constant. The blocks are read in the order of one of the sentences. These long jumps are combined with the "classical" Levenshtein edit operations, namely *insertion*, *deletion*, *substitution*, and the zero-cost operation *identity*. The resulting *long jump distance* d_{LJ} gives the minimum number of operations which are necessary to transform the candidate sentence into the reference sentence. Like the Levenshtein distance, the long jump distance can be depicted using an alignment grid as shown in Figure 1: Here, each grid point corresponds to a pair of inter-word positions in candidate and reference sentence, respectively. d_{LJ} is the minimum cost of a path between the lower left (first) and the upper right (last) alignment grid point which covers all reference and candidate words. Deletions and insertions correspond to horizontal and vertical edges, respectively. Substitutions and identity operations correspond to diagonal edges. Edges between arbitrary grid points from the same row correspond to long jump operations. It is easy to see that d_{LJ} is symmetrical.

Lopresti and Tomkins (1997) showed that finding an optimal path in a long jump alignment grid is an NP-hard problem. Our experiments showed that the calculation of exact long jump distances thus becomes impractical for sentences longer than about 20 words.

A possible way to achieve polynomial run-time is to restrict the number of admissible block permutations, for example as in Section 4. Alternatively, a heuristic or approximative distance can be calculated, as in GTM (Turian et al., 2003). An implementation of both approaches at the same time can be found in TER. In the following section we will present another approach which has a suitable run-time, while still maintaining completeness of the calculated measure. The idea of the proposed method is to drop some restrictions on the alignment path.

The long jump distance as well as the Levenshtein distance require both reference and candidate translation to be covered *completely* and *disjointly*. When extending the metric by block movements, we drop this constraint for the candidate translation. That is, only the words in the reference sentence have to be covered exactly once, whereas those in the candidate sentence can be covered zero, one, or multiple times. Dropping the constraints allows for an efficient computation of the distance. We drop the constraints for the candidate sentence and not for the reference sentence because we do not want any information contained in the reference to be omitted. Moreover, the reference translation will not contain unnecessary repetitions of blocks.

The new measure, which we call CDER, can thus be seen as a measure oriented towards *recall*, while measures like BLEU are guided by *precision*. The CDER is based on the $\overline{CD}CD$ *distance*[1] introduced by Lopresti and Tomkins (1997). The authors show that the problem of finding the optimal solution can be solved in $O(I^2 \cdot J)$ time, where I is the length of the candidate sentence and J the length of the reference sentence. Within this paper, we will refer to this distance as d_{CD} . In (Leusch et al., 2006) we showed how it can be computed in $O(I \cdot J)$ time using a modification of the Levenshtein algorithm.

We also studied the reverse direction of the described measure; that is, we dropped the coverage constraints for the reference sentence instead of the candidate sentence. Additionally, the maximum of both directions has been considered as distance measure.

4 INVWER

Another approach to circumvent the NP-hardness of the block reordering problem is to reduce the search space by restricting the number of admissible block permutations:

4.1 Bracketed transpositions

In order to reduce the complexity of the search, we restrict consequent block transpositions to be *bracketed*, i.e. the two blocks to be swapped must both lie either completely within or completely out of any blocks from previous operations. The following examples illustrate admissible and forbidden block transpositions. The brackets indicate the blocks that are swapped. In the transformation of $ABCD$ into $CDBA$ in (1), only transpositions within these blocks are performed. In (2), the transformation from $BCDA$ into $BDAC$ crosses the blocks BCD and A from the previous transposition and is therefore forbidden.

1. Admissible transpositions:
   (A)(B C D) $\rightarrow$ ((B) (C D))(A)
   $\rightarrow$ ((C D) (B))(A)
2. Forbidden transpositions:
   (A)(B C D) $\rightarrow$ (B C D)(A)
   $\not\rightarrow$ (B)(D A)(C)

A concise definition of the Levenshtein and block transposition (shift) edit operations can be given using bracketing transduction grammars.

4.2 Bracketing Transduction Grammars

A bracketing transduction grammar (BTG) (Wu, 1995) is a pair-of-strings 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 symbols, 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 $[\cdot]$, or *inverted*, denoted by $\langle\cdot\rangle$. 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 nonterminal symbol A, and each production rule consists of either a string of As or a terminal pair.

Using the BTG formalism, we can describe the edit operations Inversion *(= Shift)*, Substitution, Deletion, Insertion, as production rules, associated with a cost function c:

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

We define the *inversion edit distance* between a candidate sentence e_1^I and a reference sentence $\tilde{e}_1^J$ to be the minimum cost of the set $T(s_1^I, t_1^J)$ of all parse

[1] C stands for *cover* and D for *disjoint*. We adopted this notion for our measures.

trees generated by the BTG for this sentence pair:

$$d_{inv}(s_1^I, t_1^J) := \min_{\tau \in T(s_1^I, t_1^J)} c(\tau) \tag{1}$$

Note that, without the inversion rule, the minimum production cost equals the Levenshtein distance.

We use this distance to define our error measure, the *Inversion Word Error Rate* (INVWER), by normalizing it by the reference length. The distance can be calculated by an algorithm similar to a 2-dimensional CYK algorithm in time $O(I^3 J^3)$ and space $O(I^2 J^2)$, as described in (Leusch et al., 2003). Because the algorithm has basically a time complexity in $\Theta(I^6)$ if $I \approx J$, it can become quite slow for long sentences. Because of this, we split sentences longer than 30 words, parallel in candidate and reference on PER-optimal split points.

5 Word-dependent Substitution Costs

All automatic error measures which are based on the edit distance (for example WER, PER, TER, CDER, INVWER) assume fixed costs for the substitution of words. However, this is counter-intuitive, as replacing a word with another one which has a similar meaning will rarely change the meaning of a sentence significantly. On the other hand, 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. METEOR (Banerjee and Lavie, 2005) uses a similar idea of graduated similarity between words (exact match, stem match, WORDNET match), but instead of assuming different costs, it uses a matching procedure which matches the most similar words first. The MT system combination approach of Ayan et al. (2008) uses WORDNET matches as well as exact matches, and uses different costs for these matches.

For algorithmic reasons, it is helpful to demand that an arbitrary substitution cost function c_{SUB} for two words e, $\tilde{e}$ meets the following requirements:

1. c_{SUB} depends only on e and $\tilde{e}$.

2. c_{SUB} is a metric; especially

   (a) The costs are zero if $e = \tilde{e}$, and larger than zero otherwise.

   (b) The triangular inequation holds: it is always cheaper to replace e by $\tilde{e}$ than to replace e by e' and then e' by $\tilde{e}$.

3. The costs of substituting a word e by $\tilde{e}$ are always equal or lower than those of deleting e and then inserting $\tilde{e}$. In short, $c_{\text{SUB}} \leq 2$.

Under these conditions the algorithms for WER and CDER can easily be modified to use word-dependent substitution costs. For example, the only necessary modification in the CDER algorithm is to replace c_{SUB} by $c_{\text{SUB}}(e, \tilde{e})$ in Subsection 4.2.

For PER, it is no longer possible to use a linear time algorithm in the general case. Instead, we use a modification of the Hungarian algorithm (Knuth, 1993).

The question is now how to define the word-dependent substitution costs. 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 we consider the words to be, and the lower we want the substitution costs between them to be. In English, this works well with similar tenses of the same verb, or with genitives or plurals of the same noun. Nevertheless, a similar spelling is no guarantee for a similar meaning, because prefixes such as “mis-”, “in-”, or “un-” can change the meaning of a word significantly.

We have studied two different approaches to use the similarity in the spelling of two words as a substitution cost:

5.1 Character-based Levenshtein Distance

An obvious way of comparing the spelling is the Levenshtein distance. Here, words are compared on character level. To normalize this distance into a range from 0 (for identical words) to 1 (for completely different words), we divide the absolute distance by the length of the Levenshtein alignment path.

5.2 Common Prefix Length

Another character-based substitution cost function we studied is based on the common prefix length of both words. In English, different tenses of the same verb share the same prefix; which is usually the stem. The same holds for different cases, numbers

Table 1: Example of word-dependent substitution costs.

		Levenshtein		prefix	
e	$\tilde{e}$	distance	substitution cost	similarity	substitution cost
usual	unusual	2	$\frac{2}{7} = 0.29$	1	$1 - \frac{1}{6} = 0.83$
understanding	misunderstanding	3	$\frac{3}{16} = 0.19$	0	1.00
talk	talks	1	$\frac{1}{5} = 0.20$	4	$1 - \frac{4}{4.5} = 0.11$

and genders of most nouns and adjectives. However, it does not hold if verb prefixes are changed or removed. On the other hand, the common prefix length is sensitive to critical prefixes such as "mis-" for the same reason. Consequently, the common prefix length, normalized by the average length of both words, gives a reasonable measure for the similarity of two words. To transform the normalized common prefix length into costs, this fraction is then subtracted from 1. An example for these two approaches is shown in Table 1.

6 Linear Combination of evaluation measures

An interesting topic in MT evaluation research is the question whether different MT evaluation measures can be combined into a consensus score, which hopefully shows a better correlation with the target – human evaluation – than the single measures. Recently, Albrecht & Hwa (2007) have investigated on the combination of up to 53 measures and features in a regression model and a classifier as evaluation measure. Also, a linear combination of BLEU and TER has been successfully used for tuning MT systems (Mauser et al., 2008; Rosti et al., 2008). In our approach, we only are interested in the linear combination of two MT evaluation measures, particularly the combination of CDER and PER. We expect this combination to have a higher correlation with human evaluation than the measures alone. CDER (as opposed to PER) has the ability to reward correct local ordering, whereas PER (as opposed to CDER) penalizes overly long candidate sentences. The two measures were combined with linear interpolation. In order to determine the weights, we performed data analysis on seven different corpora in (Leusch et al., 2006). The results were consistent across all different data collections and language pairs: a linear combination of about 60% CDER and 40% PER has a significantly higher correlation with human evaluation than each of the measures alone. Consequently, we chose these weights for the NIST Metrics MATR evaluation as well.

Table 2: Corpus statistics of the MATR MT06 corpus that was used for experimental evaluation of the proposed measures.

Language pair	(Arabic)–English
Genre	Newswire texts
MT systems	8
Documents	25
Segments	249
References/seg	4
Hyp. length	32.5
Ref. length	34.3
Human evaluation	adequacy (1. . . 7)

7 Experimental results

For an experimental comparison of the different evaluation measures, we calculated the correlation between these measures and human evaluation, in particular the "adequacy", for the MT06 corpus provided by NIST and LDC for the Metrics MATR 2008 evaluation (NIST, 2008). This corpus consists of translations of 25 Arabic newswire documents into English, as generated by 8 MT systems that participated in NIST's 2006 MT evaluation. Some statistics on this corpus are listed in Table 2.

Within NIST's Metrics MATR evaluation, another corpus was provided to participants for metrics evaluation. But due to its extremely limited size – a single document consisting of 16 segments – correlation results generated on this corpus are quite noisy, and of limited significance.

Table 3: Pearson's r and Kendall's τ (absolute) between adequacy and automatic evaluation measures on different levels of the MATR MT06 data.

Measure	r			τ			$\bar{\tau}$	
	seg	doc	sys	seg	doc	sys	sys/seg	sys/doc
WER	0.621	0.853	0.953	0.503	0.599	0.571	0.584	0.641
WER $+c(w)$	0.626	0.860	0.954	0.506	0.608	0.571	0.580	0.653
PER	0.597	0.852	0.958	0.482	0.588	0.643	0.576	0.644
PER $+c(w)$	0.586	0.858	0.966	0.483	0.590	0.714	0.559	0.646
BLEU (min Ref)	0.592	0.844	0.943	0.476	0.580	0.571	0.578	0.588
BLEU (avg Ref)	0.598	0.857	0.955	0.483	0.602	0.643	0.587	0.612
BLEUS	0.672	0.860	0.955	0.541	0.606	0.643	0.590	0.615
BLEUSP	0.687	0.877	0.961	0.542	0.613	0.643	0.609	0.618
TER	0.597	0.849	0.957	0.495	0.602	0.571	0.595	0.667
INVWER	0.638	0.867	0.958	0.509	0.606	0.571	0.583	0.643
INVWER $+c(w)$	0.649	0.871	0.958	0.512	0.606	0.571	0.573	0.638
CDER $+c(w)$	0.708	0.891	0.973	0.558	0.643	0.714	0.623	0.671
CDER + PER $+c(w)$	0.690	0.885	0.975	0.543	0.632	0.714	0.610	0.679

$+c(w)$ denotes measures using word dependent substitution costs.
“seg” is on segment level, *“doc* is on document level, *“sys”* is on system level.
“sys/seg” is average system ranking per segment, *“sys/doc”* is average system ranking per document.

We investigated two different aspects of automatic evaluation measures – their ability to give a reliable absolute estimate of their translation quality, and their ability to rank different translations with regard to their quality. As target for this, we chose the adequacy score, an integer value between 1 and 7, that was assigned to all translations in the corpus by a human judge. We measured the absolute prediction as Pearson's correlation coefficient r (Casella and Berger, 1990), and the ranking capability as Kendall's τ (Kendall, 1970). The latter has the big advantage over other coefficients like Spearman's ρ that it handles ties in a well-defined and reasonable manner. As there are only seven different outcomes for adequacy, but up to several thousand samples, ties are extremely frequent in these experiments.

We measured both r and τ on three levels of granularity – the system level, that is comparing the accumulated scores of all documents per system, the document level, and the segment level.

Comparing the scores of different systems at the same time as comparing different (source) documents or even segments raises the problem that the measure is used at the same time to compare the output of different MT systems, and to compare the output for different source sentence. In other words, an evaluation metric gets a “bonus” in terms of correlation already if it is able to divide easy from difficult source sentences (which typically have good and bad translations respectively), as well as for dividing different MT systems. In practice, we are mostly interested in the latter: Our test corpus then is fixed, and there might be more efficient methods for estimating the “difficulty” of a source sentence. We use the MT evaluation measure here because we want to compare the actual MT systems, and not the source sentences, and the MT evaluation measure we chose should respect this.

To divide these two effects in our experiments, we calculated the rank correlation (using τ) over the different MT system for a fixed source segment or document respectively, and then calculated the arithmetic average over the different τ for the individual source segments (or documents).[2] We denote this averaged correlation coefficient as $\bar{\tau}$. Our experimental results are listed in Table 3.

[2] This is another advantage of τ over r or ρ – in our understanding, it seems at least questionable whether calculating the arithmetic average would be a valid procedure for the latter two.

7.1 BLEU and BLEUSP

Using the average reference length instead of the minimum reference length brings a very small improvement in correlation with human judgment on segment level, and a slightly larger improvement on document and system level. The largest improvement for BLEU-like measures on the segment level comes from Lin & Och's smoothed geometric mean in BLEUS: r raises from .60 to .67, τ from .48 to .54, even though the ability to rank systems on the sentence or document level hardly increases. This can be increased using the segment boundary tokens: r improves from .67 to .68, and $\bar{\tau}_{seg}$ from .59 to .61.

7.2 TER and INVWER

Even though TER and INVWER are structurally very similar, we see that INVWER has a significantly higher correlation than TER with human evaluation on segment and document level, even though it seems that TER has a better capability to actually judge between MT systems on smaller levels. Using word-dependent substitution costs brings small improvements on the segment level, but seems to have a slightly negative effect on the ability to differentiate between systems.

7.3 Word-dependent substitution costs

Compared with our results in (Leusch et al., 2006), we see only small improvements, or for PER even a very small deterioration if we use word-dependent substitution costs on this corpus, both on segment and system level. For our final submission, we made the substitution costs dependent on the common prefix length in out Metrics MATR submission.

7.4 CDER and linear combination of evaluation measures

CDER, as a pure recall measure, shows again the highest correlation of all evaluation measures, on segment, document, and system level. Unfortunately, it is unwise to use a purely recall-oriented measure in actual research. This is because MT systems tuned for such a measure, either directly or indirectly, tend to produce overly long sentences containing many unnecessary or even wrong insertions. To avoid this, and because doing so showed a significant increase in correlation with human judgment on most of our development corpora, we use a linear combination of CDER (with a weight of .6) and PER (.4), both using word dependent substitution costs. Unfortunately, this combined measure shows a slightly lower correlation with human judgment on the sentence level than pure CDER. But even though, this measure has a significantly higher correlation on all levels than the original BLEU or TER score, and is better than or on par with even our improved scores INVWER and BLEUSP.

8 Conclusions

In this paper we have presented three improved evaluation measures for MT, based on the well-established and understood measures TER, BLEU, and a variant of WER and PER. While not exactly being revolutionary, our measures show a significant improvement in correlation with human judgment compared with the original measures. We further refined the way this correlation is measured, taking into account typical statistical and data effects when looking evaluating MT evaluation measures. We consider our results to be of importance because a multitude of new, sometimes "revolutionary" MT evaluation measures have been proposed over the last years, sometimes to be compared only with BLEU or TER in terms of correlation, even for applications where these baseline measures are not well suited for. One of our scientific contributions in this paper is to show that even with these only alterations and additions, we can have a significantly higher correlation with human judgment. Along the way, we hope to raise the baseline for new evaluation measures significantly.

The measures described in this paper, as well as some additional measures, have been implemented by us under a `python` command line tool called `PyET`. The implementation uses shared libraries written in `C++` for performance. Please contact the first author via E-Mail, or visit his web site[3] to obtain a copy of this software.

9 Acknowledgments

This paper is based upon work supported by the Defense Advanced Research Projects Agency

[3] http://www-i6.informatik.rwth-aachen.de/~leusch/

(DARPA) under Contract No. HR0011-06-C-0023.

References

J. Albrecht and R. Hwa. 2007. A re-examination of machine learning approaches for sentence-level mt evaluation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pp. 880–887, Prague, Czech Republic, June. Association for Computational Linguistics.

N. F. Ayan, J. Zheng, and W. Wang. 2008. Improving alignments for better confusion networks for combining machine translation systems. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pp. 33–40, Manchester, UK, August. Coling 2008 Organizing Committee.

S. Banerjee and A. Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pp. 65–72, Ann Arbor, Michigan, June. Association for Computational Linguistics.

G. Casella and R. L. Berger, 1990. *Statistical Inference*, chapter 4.5, pp. 160–168. Duxbury Press.

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

D. E. Knuth, 1993. *The Stanford GraphBase: a platform for combinatorial computing*, pp. 74–87. ACM Press, New York, NY.

G. Leusch, N. Ueffing, and H. Ney. 2003. A novel string-to-string distance measure with applications to machine translation evaluation. In *Proc. MT Summit IX*, pp. 240–247, New Orleans, LA, September.

G. Leusch, N. Ueffing, D. Vilar, and H. Ney. 2005. Preprocessing and normalization for automatic evaluation of machine translation. In *43rd Annual Meeting of the Assoc. for Computational Linguistics: Proc. Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization*, pp. 17–24, Ann Arbor, Michigan, June. Association for Computational Linguistics.

G. Leusch, N. Ueffing, and H. Ney. 2006. CDER: Efficient mt evaluation using block movements. In *Conference of the European Chapter of the Association for Computational Linguistics*, pp. 241–248, Trento, Italy, April. European Chapter of the Association for Computational Linguistics.

V. I. Levenshtein. 1966. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, 10(8):707–710, Feb.

C. Y. Lin and F. J. Och. 2004. Orange: a method for evaluation automatic evaluation metrics for machine translation. In *Proc. COLING 2004*, pp. 501–507, Geneva, Switzerland, August.

D. Lopresti and A. Tomkins. 1997. Block edit models for approximate string matching. *Theoretical Computer Science*, 181(1):159–179, July.

A. Mauser, S. Hasan, and H. Ney. 2008. Automatic evaluation measures for statistical machine translation system optimization. In *International Conference on Language Resources and Evaluation*, Marrakech, Morocco, May.

NIST. 2008. NIST Metrics for Machine Translation Challenge 2008 . `http://www.nist.gov/speech/tests/metricsmatr/2008/` .

K. A. Papineni, S. Roukos, T. Ward, and W. J. Zhu. 2001. Bleu: a method for automatic evaluation of machine translation. Technical Report RC22176 (W0109-022), IBM Research Division, Thomas J. Watson Research Center, September.

A. V. Rosti, S. Matsoukas, and R. Schwartz. 2007. Improved word-level system combination for machine translation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pp. 312–319, Prague, Czech Republic, June. Association for Computational Linguistics.

A. V. Rosti, B. Zhang, S. Matsoukas, and R. Schwartz. 2008. Incremental hypothesis alignment for building confusion networks with application to machine translation system combination. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pp. 183–186, Columbus, Ohio, June. Association for Computational Linguistics.

M. Snover, B. J. Dorr, R. Schwartz, J. Makhoul, L. Micciulla, and R. Weischedel. 2005. A study of translation error rate with targeted human annotation. Technical Report LAMP-TR-126, CS-TR-4755, UMIACS-TR-2005-58, University of Maryland, College Park, MD.

J. P. Turian, L. Shen, and I. D. Melamed. 2003. Evaluation of machine translation and its evaluation. In *Proc. MT Summit IX*, pp. 23–28, New Orleans, LA, September.

D. Wu. 1995. An algorithm for simultaneously bracketing parallel texts by aligning words. In *Proc. of the 33rd Annual Conf. of the Association for Computational Linguistics*, pp. 244–251.