LRscore for Evaluating Lexical and Reordering Quality in MT

Alexandra Birch
University of Edinburgh
United Kingdom
a.c.birch-mayne@s0454866.ed.ac.uk

Miles Osborne
University of Edinburgh
United Kingdom
miles@inf.ed.ac.uk

Abstract

The ability to measure the quality of word order in translations is an important goal for research in machine translation. Current machine translation metrics do not adequately measure the reordering performance of translation systems. We present a novel metric, the LRscore, which directly measures reordering success. The reordering component is balanced by a lexical metric. Capturing the two most important elements of translation success in a simple combined metric with only one parameter results in an intuitive, shallow, language independent metric.

1 Introduction

The main purpose of MT evaluation is to determine "to what extent the makers of a system have succeeded in mimicking the human translator" (Krauwer, 1993). But machine translation has no "ground truth" as there are many possible correct translations. It is impossible to judge whether a translation is incorrect or simply unknown and it is even harder to judge the the degree to which it is incorrect. Even so, automatic metrics are necessary. It is nearly impossible to collect enough human judgments for evaluating incremental improvements in research systems, or for tuning statistical machine translation system parameters. Automatic metrics are also much faster and cheaper than human evaluation and they produce reproducible results.

Machine translation research relies heavily upon automatic metrics to evaluate the performance of models. However, current metrics rely upon indirect methods for measuring the quality of the word order, and their ability to capture reordering performance has been demonstrated to be poor (Birch et al., 2010). There are two main approaches to capturing reordering. The first way to measure the quality of word order is to count the number of matching n-grams between the reference and the hypothesis. This is the approach taken by the BLEU score (Papineni et al., 2002). This method discounts any n-gram which is not identical to a reference n-gram, and also does not consider the relative position of the strings. They can be anywhere in the sentence. Another common approach is typified by METEOR (Banerjee and Lavie, 2005) and TER (Snover et al., 2006). They calculate an ordering penalty for a hypothesis based on the minimum number of chunks the translation needs to be broken into in order to align it to the reference. The disadvantage of the second approach is that aligning sentences with very different words can be inaccurate. Also there is no notion of how far these blocks are out of order. More sophisticated metrics, such as the RTE metric (Padó et al., 2009), use higher level syntactic or even semantic analysis to determine the quality of the translation. These approaches are useful, but can be very slow, require annotation, they are language dependent and their parameters are hard to train. For most research work shallow metrics are more appropriate.

Apart from failing to capture reordering performance, another common criticism of most current automatic MT metrics is that a particular score value reported does not give insights into quality (Przybocki et al., 2009). This is because there is no intrinsic significance of a difference in scores. Ideally, the scores that the metrics report would be meaningful and stand on their own. However, the most one can say is that higher is better for accuracy metrics and lower is better for error metrics.

We present a novel metric, the LRscore, which explicitly measures the quality of word order in machine translations. It then combines the reordering metric with a metric measuring lexical success. This results in a comprehensive met-

ric which measures the two most fundamental aspects of translation. We argue that the LRscore is intuitive and meaningful because it is a simple, decomposable metric with only one parameter to train.

The LRscore has many of the properties that are deemed to be desirable in a recent metric evaluation campaign (Przybocki et al., 2009). The LRscore is language independent. The reordering component relies on abstract alignments and word positions and not on words at all. The lexical component of the system can be any meaningful metric for a particular target language. In our experiments we use 1-gram BLEU and 4-gram BLEU, however, if a researcher was interested in morphologically rich languages, a different metric which scores partially correct words might be more appropriate. The LRscore is a shallow metric, which means that it is reasonably fast to run. This is important in order to be useful for training of the translation model parameters. A final advantage is that the LRscore is a sentence level metric. This means that human judgments can be directly compared to system scores and helps researchers to understand what changes they are seeing between systems.

In this paper we start by describing the reordering metrics and then we present the LRscore. Finally we discuss related work and conclude.

2 Reordering Metrics

The relative ordering of words in the source and target sentences is encoded in alignments. We can interpret alignments as permutations. This allows us to apply research into metrics for ordered encodings to our primary tasks of measuring and evaluating reorderings. A word alignment over a sentence pair allows us to transcribe the source word positions in the order of the aligned target words. Permutations have already been used to describe reorderings (Eisner and Tromble, 2006), primarily to develop a reordering model which uses ordering costs to score possible permutations. Here we use permutations to evaluate reordering performance based on the methods presented in (Birch et al., 2010).

The ordering of the words in the target sentence can be seen as a permutation of the words in the source sentence. The source sentence s of length N consists of the word positions $s_0 \cdots s_i \cdots s_N$. Using an alignment function where a source word at position i is mapped to a target word at position j with the function $a : i \rightarrow j$, we can reorder the source word positions to reflect the order of the words in the target. This gives us a permutation.

A **permutation** is a bijective function from a set of natural numbers $1, 2, \cdots, N$ to itself. We will name our permutations π and σ. The i^{th} symbol of a permutation π will be denoted as $\pi(i)$, and the inverse of the permutation π^{-1} is defined so that if $\pi(i) = j$ then $\pi^{-1}(j) = i$. The identity, or monotone, permutation id is the permutation for which $id(i) = i$ for all i. Table 1 shows the permutations associated with the example alignments in Figure 1. The permutations are calculated by iterating over the source words, and recording the ordering of the aligned target words.

Permutations encode one-one relations, whereas alignments contain null alignments and one-many, many-one and many-many relations. For now, we make some simplifying assumptions to allow us to work with permutations. Source words aligned to null ($a(i) \rightarrow null$) are assigned the target word position immediately after the target word position of the previous source word ($\pi(i) = \pi(i-1) + 1$). Where multiple source words are aligned to the same target word or phrase, a many-to-one relation, the target ordering is assumed to be monotone. When one source word is aligned to multiple target words, a one-to-many relation, the source word is assumed to be aligned to the first target word.

A translation can potentially have many valid word orderings. However, we can be reasonably certain that the ordering of reference sentence must be acceptable. We therefore compare the ordering of a translation with that of the reference sentence. The underlying assumption is that most reasonable word orderings should be fairly similar to the reference. The assumption that the reference is somehow similar to the translation is necessary for all automatic machine translation metrics. We propose using permutation distance metrics to perform the comparison.

There are many different ways of measuring distance between two permutations, with different solutions originating in different domains (statistics, computer science, molecular biology, ...). Real numbered data leads to measures such as Euclidean distance, binary data to measures such as Hamming distance. But for ordered sets, there are many different options, and the best one de-

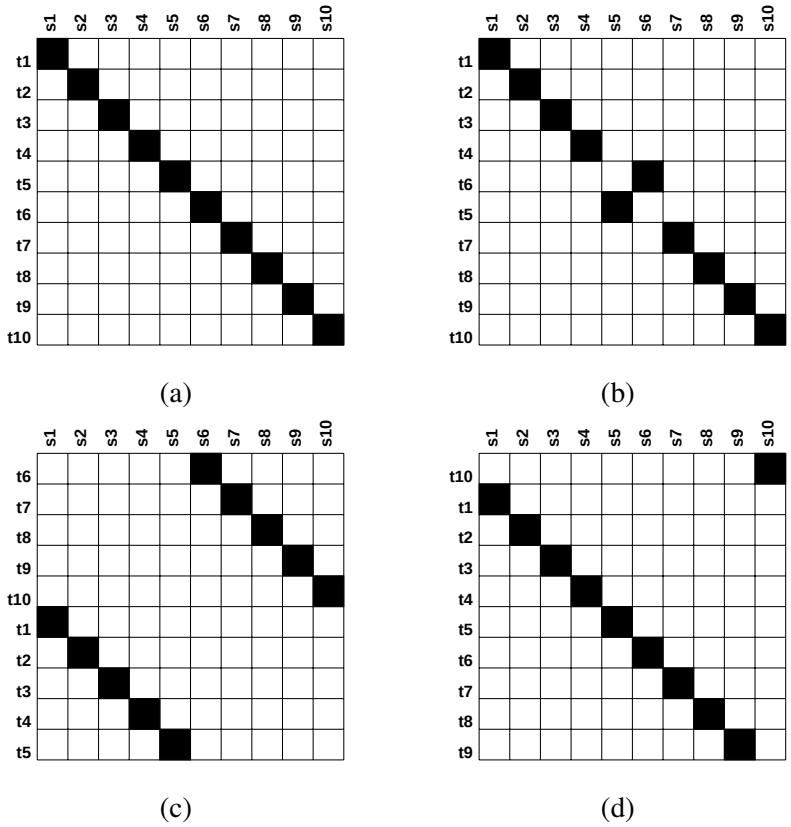

Figure 1: Synthetic examples: a translation and three reference scenarios. (a) is a monotone translation, (b) is a reference with one short distance word order difference, (c) is a reference where the order of the two halves has been swapped, and (d) is a reference with a long distance reordering of the first target word.

pends on the task at hand. We choose a few metrics which are widely used, efficient to calculate and capture certain properties of the reordering. In particular, they are sensitive to the number of words that are out of order. Three of the metrics, Kendall's tau, Spearman's rho and Spearman's footrule distances also take into account the distance between positions in the reference and translation sentences, or the size of the reordering.

An obvious disadvantage of this approach is the fact that we need alignments, either between the source and the reference, and the source and the translation, or directly between the reference and the translation. If accuracy is paramount, the test set could include manual alignments and the systems could directly output the source-translation alignments. Outputting the alignment information should require a trivial change to the decoder. Alignments can also be automatically generated using the alignment model that aligns the training data.

Distance metrics increase as the quality of translation decreases. We invert the scale of the distance metrics in order to easily compare them with other metrics where increases in the metrics mean increases in translation quality. All permutation distance metrics are thus subtracted from 1. Note that the two permutations we refer to π and σ are relative to the source sentence, and not to the reference: the source-reference permutation is compared to the source-translation permutation.

(a)	(1 2 3 4 5 6 7 8 9 10)
(b)	(1 2 3 4 •6 •5 •7 8 9 10)
(c)	(6 7 8 9 10 •1 2 3 4 5)
(d)	(2 3 4 5 6 7 8 9 10 •1)

Table 1: Permutations extracted from the sentence pairs shown in Figure 1: (a) is a monotone permutation and (b), (c) and (d) are permutations with different amounts of disorder, where bullet points highlight non-sequential neighbors.

2.1 Hamming Distance

The Hamming distance (Hamming, 1950) measures the number of disagreements between two

permutations. The Hamming distance for permutations was proposed by (Ronald, 1998) and is also known as the **exact match distance**. It is defined as follows:

$$d_H(\pi,\sigma) = 1 - \frac{\sum_{i=1}^{n} x_i}{n} \text{ where } x_i = \begin{cases} 0 \text{ if } \pi(i) = \sigma(i) \\ 1 \text{ otherwise} \end{cases}$$

Where π, σ are the two permutations and the normalization constant Z is n, the length of the permutation. We are interested in the Hamming distance for its ability to capture the amount of absolute disorder that exists between two permutations. The Hamming distance is widely utilized in coding theory to measure the discrepancy between two binary sequences.

2.2 Kendall's Tau Distance

Kendall's tau distance is the minimum number of transpositions of two *adjacent* symbols necessary to transform one permutation into another (Kendall, 1938; Kendall and Gibbons, 1990). This is sometimes known as the **swap distance** or the **inversion distance** and can be interpreted as a function of the probability of observing concordant and discordant pairs (Kerridge, 1975). It is defined as follows:

$$\begin{aligned} d_\tau(\pi,\sigma) &= 1 - \frac{\sum_{i=1}^{n}\sum_{j=1}^{n} z_{ij}}{Z} \\ \text{where } z_{ij} &= \begin{cases} 1 \text{ if } \pi(i) < \pi(j) \text{ and } \sigma(i) > \sigma(j) \\ 0 \text{ otherwise} \end{cases} \\ Z &= \frac{(n^2 - n)}{2} \end{aligned}$$

The Kendall's tau metric is possibly the most interesting for measuring reordering as it is sensitive to all relative orderings. It consequently measures not only how many reordering there are but also the distance that words are reordered.

In statistics, Spearman's rho and Kendall's tau are widely used non-parametric measures of association for two rankings. In natural language processing research, Kendall's tau has been used as a means of estimating the distance between a system-generated and a human-generated gold-standard order for the sentence ordering task (Lapata, 2003). Kendall's tau has also been used in machine translation as a cost function in a reordering model (Eisner and Tromble, 2006) and an MT metric called ROUGE-S (Lin and Och, 2004) is similar to a Kendall's tau metric on lexical items. ROUGE-S is an F-measure of ordered pairs of words in the translation. As far as we know, Kendall's tau has not been used as a reordering metric before.

3 LRscore

The goal of much machine translation research is either to improve the quality of the words used in the output, or their ordering. We use the reordering metrics and combine them with a measurement of lexical performance to produce a comprehensive metric, the LRscore. The LRscore is a linear interpolation of a reordering metric with the BLEU score. If we use the 1-gram BLEU score, BLEU1, then the LRscore relies purely upon the reordering metric for all word ordering evaluation. We also use the 4-gram BLEU score, BLEU4, as it is an important baseline and the values it reports are very familiar to machine translation researchers. BLEU4 also contains a notion of word ordering based on longer matching n-grams. However, it is aware only of very local orderings. It does not measure the magnitude of the orderings like the reordering metrics do, and it is dependent on exact lexical overlap which does not affect the reordering metric. The two components are therefore largely orthogonal and there is a benefit in combining them. Both the BLEU score and the reordering distance metric apply a brevity penalty to account for translations of different lengths.

The formula for calculating the LRscore is as follows:

$$LRscore = \alpha * R + (1 - \alpha) BLEU$$

Where the reordering metric R is calculated as follows:

$$R = d * BP$$

Where we either take the Hamming distance d_H or the Kendall's tau distance d_τ as the reordering distance d and then we apply the brevity penalty BP. The brevity penalty is calculated as:

$$BP = \begin{cases} 1 & \text{if } t > r \\ e^{1-r/t} & \text{if } t \leq r \end{cases}$$

where t is the length of the translation, and r is the closest reference length. R is calculated at the sentence level, and the scores are averaged over a test set. This average is then combined with the

system level lexical score. The Lexical metric is the BLEU score which sums the log precision of n-grams. In our paper we set the n-gram length to either be one or four.

The only parameter in the metric α balances the contribution of reordering and the lexical components. There is no analytic solution for optimizing this parameter, and we use greedy hillclimbing in order to find the optimal setting. We optimize the sentence level correlation of the metric to human judgments of accuracy as provided by the WMT 2010 shared task. As hillclimbing can end up in a local minima, we perform 20 random restarts, and retaining only the parameter value with the best consistency result. Random-restart hill climbing is a surprisingly effective algorithm in many cases. It turns out that it is often better to spend CPU time exploring the space, rather than carefully optimizing from an initial condition.

The brevity penalty applies to both the reordering metric and the BLEU score. We do not set a parameter to regulate the impact of the brevity penalty, as we want to retain BLEU scores that are comparable with BLEU scores computed in published research. And as we do not regulate the brevity penalty in the BLEU score, we do not wish to do so for the reordering metric either. It therefore impacts on both the reordering and the lexical components equally.

4 Correlation with Human Judgments

It has been common to use seven-point fluency and adequacy scores as the main human evaluation task. These scores are intended to be absolute scores and comparable across sentences. Seven-point fluency and adequacy judgements are quite unreliable at a sentence level and so it seems dubious that they would be reliable across sentences. However, having absolute scores does have the advantage of making it easy to calculate the correlation coefficients of the metric with human judgements. Using rank judgements, we do not have absolute scores and thus we cannot compare translations across different sentences.

We therefore take the method adopted in the 2009 workshop on machine translation (Callison-Burch et al., 2009). We ascertained how consistent the automatic metrics were with the human judgements by calculating consistency in the following manner. We take each pairwise comparison of translation output for single sentences by a particular judge, and we recorded whether or not the metrics were consistent with the human rank. Ie. we counted cases where both the metric and the human judged agree that one system is better than another. We divided this by the total umber of pairwise comparisons to get a percentage. There were many ties in the human data, but metrics rarely give the same score to two different translations. We therefore excluded pairs that the human annotators ranked as ties. The human ranking data and the system outputs from the 2009 Workshop on Machine Translation (Callison-Burch et al., 2009) have been used to evaluate the LRscore.

Metric	de-en	es-en	fr-en	cz-en
BLEU4	58.72	55.48	57.71	57.24
LR-HB1	60.37	**60.55**	58.59	53.70
LR-HB4	60.49	58.88	**58.80**	57.74
LR-KB1	60.67	58.54	58.46	54.20
LR-KB4	**61.07**	59.86	58.59	**58.92**

Table 2: The percentage consistency between human judgements of rank and metrics. The LRscore variations (LR-*) are optimised for consistency for each language pair.

We optimise the sentence level consistency of the metric. As hillclimbing can end up in a local minima, we perform 20 random restarts, and retaining only the parameter value with the best consistency result. Random-restart hill climbing is a surprisingly effective algorithm in many cases. It turns out that it is often better to spend CPU time exploring the space, rather than carefully optimising from an initial condition.

Table 2 reports the optimal consistency of the LRscore and baseline metrics with human judgements for each language pair. The table also reports the individual component results. The LRscore variations are named as follows: LR refers to the LRscore, "H" refers to the Hamming distance and "K" to Kendall's tau distance. "B1" and "B4" refer to the smoothed BLEU score with the 1-gram and 4-gram scores. The LRscore is the metric which is most consistent with human judgement. This is an important result which shows that combining lexical and reordering information makes for a stronger metric.

5 Related Work

(Wong and Kit, 2009) also suggest a metric which combines a word choice and a word order com-

ponent. They propose a type of F-measure which uses a matching function M to calculate precision and recall. M combines the number of matched words, weighted by their *tfidf* importance, with their position difference score, and finally subtracting a score for unmatched words. Including unmatched words in the in M function undermines the interpretation of the supposed F-measure. The reordering component is the average difference of absolute and relative word positions which has no clear meaning. This score is not intuitive or easily decomposable and it is more similar to METEOR, with synonym and stem functionality mixed with a reordering penalty, than to our metric.

6 Conclusion

We propose the LRscore which combines a lexical and a reordering metric. This results in a metric which is both meaningful and accurately measures the word order performance of the translation model.

References

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for MT evaluation with improved correlation with human judgments. In *Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization*.

Alexandra Birch, Phil Blunsom, and Miles Osborne. 2010. Metrics for MT Evaluation: Evaluating Reordering. *Machine Translation (to appear)*.

Chris Callison-Burch, Philipp Koehn, Christof Monz, and Josh Schroeder. 2009. Findings of the 2009 Workshop on Statistical Machine Translation. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 1–28, Athens, Greece, March. Association for Computational Linguistics.

Jason Eisner and Roy W. Tromble. 2006. Local search with very large-scale neighborhoods for optimal permutations in machine translation. In *Proceedings of the HLT-NAACL Workshop on Computationally Hard Problems and Joint Inference in Speech and Language Processing*, pages 57–75, New York, June.

Richard Hamming. 1950. Error detecting and error correcting codes. *Bell System Technical Journal*, 26(2):147–160.

M. Kendall and J. Dickinson Gibbons. 1990. *Rank Correlation Methods*. Oxford University Press, New York.

Maurice Kendall. 1938. A new measure of rank correlation. *Biometrika*, 30:81–89.

D Kerridge. 1975. The interpretation of rank correlations. *Applied Statistics*, 2:257–258.

S. Krauwer. 1993. Evaluation of MT systems: a programmatic view. *Machine Translation*, 8(1):59–66.

Mirella Lapata. 2003. Probabilistic text structuring: Experiments with sentence ordering. *Computational Linguistics*, 29(2):263–317.

Chin-Yew Lin and Franz Josef Och. 2004. Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04), Main Volume*, pages 605–612, Barcelona, Spain, July.

Sebastian Padó, Daniel Cer, Michel Galley, Dan Jurafsky, and Christopher D. Manning. 2009. Measuring machine translation quality as semantic equivalence: A metric based on entailment features. *Machine Translation*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the Association for Computational Linguistics*, pages 311–318, Philadelphia, USA.

Mark Przybocki, Kay Peterson, Sébastien Bronsart, and Gregory Sanders. 2009. The nist 2008 metrics for machine translation challengeoverview, methodology, metrics, and results. *Machine Translation*.

S Ronald. 1998. More distance functions for order-based encodings. In *the IEEE Conference on Evolutionary Computation*, pages 558–563.

Matthew Snover, Bonnie Dorr, R Schwartz, L Micciulla, and J Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *AMTA*.

B. Wong and C. Kit. 2009. ATEC: automatic evaluation of machine translation via word choice and word order. *Machine Translation*, pages 1–15.