Ranking vs. Regression in Machine Translation Evaluation

Kevin Duh*
Dept. of Electrical Engineering
University of Washington
Seattle, WA 98195
kevinduh@u.washington.edu

Abstract

Automatic evaluation of machine translation (MT) systems is an important research topic for the advancement of MT technology. Most automatic evaluation methods proposed to date are score-based: they compute scores that represent translation quality, and MT systems are compared on the basis of these scores.

We advocate an alternative perspective of automatic MT evaluation based on ranking. Instead of producing scores, we directly produce a ranking over the set of MT systems to be compared. This perspective is often simpler when the evaluation goal is system comparison. We argue that it is easier to elicit human judgments of ranking and develop a machine learning approach to train on rank data. We compare this ranking method to a score-based regression method on WMT07 data. Results indicate that ranking achieves higher correlation to human judgments, especially in cases where ranking-specific features are used.

1 Motivation

Automatic evaluation of machine translation (MT) systems is an important research topic for the advancement of MT technology, since automatic evaluation methods can be used to quickly determine the (approximate) quality of MT system outputs. This is useful for tuning system parameters and for comparing different techniques in cases when human judgments for each MT output are expensivie to obtain.

Many automatic evaluation methods have been proposed to date. Successful methods such as BLEU (Papineni et al., 2002) work by comparing MT output with one or more human reference translations and generating a similarity score. Methods differ by the definition of similarity. For instance, BLEU and ROUGE (Lin and Och, 2004) are based on n-gram precisions, METEOR (Banerjee and Lavie, 2005) and STM (Liu and Gildea, 2005) use word-class or structural information, Kauchak (2006) leverages on paraphrases, and TER (Snover et al., 2006) uses edit-distances. Currently, BLEU is the most popular metric; it has been shown that it correlates well with human judgments on the corpus level. However, finding a metric that correlates well with human judgments on the sentence-level is still an open challenge (Blatz and others, 2003).

Machine learning approaches have been proposed to address the problem of sentence-level evaluation. (Corston-Oliver et al., 2001) and (Kulesza and Shieber, 2004) train classifiers to discriminate between human-like translations and automatic translations, using features from the aforementioned metrics (e.g. n-gram precisions). In contrast, (Albrecht and Hwa, 2007) argues for a regression approach that directly predicts human adequecy/fluency scores.

All the above methods are score-based in the sense that they generate a score for each MT system output. When the evaluation goal is to compare multiple MT systems, scores are first generated independently for each system, then systems are ranked by their respective scores. We think that this two-step process may be unnecessarily complex. Why solve a more difficult problem of predicting the quality of MT system outputs, when the goal is simply

*Work supported by an NSF Graduate Research Fellowship.

Proceedings of the Third Workshop on Statistical Machine Translation, pages 191–194,
Columbus, June 2008.

to compare systems? In this regard, we propose a ranking-based approach that directly ranks a set of MT systems without going through the intermediary of system-specific scores. Our approach requires (a) training data in terms of human ranking judgments of MT outputs, and (b) a machine learning algorithm for learning and predicting rankings.[1]

The advantages of a ranking approach are:

- It is often easier for human judges to rank MT outputs by preference than to assign absolute scores (Vilar et al., 2007). This is because it is difficult to quantify the quality of a translation accurately, but relative easy to tell which one of several translations is better. Thus human-annotated data based on ranking may be less costly to acquire.

- The inter- and intra-annotator agreement for ranking is much more reasonable than that of scoring. For instance, Callison-Burch (2007) found the inter-annotator agreement (Kappa) for scoring fluency/adequency to be around .22-.25, whereas the Kappa for ranking is around .37-.56. Thus human-annotated data based on ranking may be more reliable to use.

- As mentioned earlier, when the final goal of the evaluation is comparing systems, ranking more directly solves the problem. A scoring approach essentially addresses a more difficult problem of estimating MT output quality.

Nevertheless, we note that score-based approaches remain important in cases when the absolute difference between MT quality is desired. For instance, one might wonder *by how much* does the top-ranked MT system outperform the second-ranked system, in which case a ranking-based approach provide no guidance.

In the following, Section 2 formulates the sentence-level MT evaluation problem as a ranking problem; Section 3 explains a machine learning approach for training and predicting rankings; this is our submission to the WMT2008 Shared Evaluation task. Ranking vs. scoring approaches are compared in Section 4.

2 Formulation of the Ranking Problem

We formulate the sentence-level MT evaluation problem as follows: Suppose there are T source sentences to be translated. Let r_t, $t = 1..T$ be the set of references[2]. Corresponding to each source sentence, there are N MT system outputs $o_t^{(n)}$, $n = 1..N$ and M_t ($M_t \leq N$) human evaluations. The evaluations are represented as M_t-dimensional label vectors y_t. In a scoring approach, the elements of y_t may correspond to, e.g. a fluency score on a scale of 1 to 5. In a ranking approach, they may correspond to relative scores that are used to represent ordering (e.g. $y_t = [6; 1; 3]$ means that there are three outputs, and the first is ranked best, followed by third, then second.)

In order to do machine learning, we extract feature vectors $x_t^{(n)}$ from each pair of r_t and $o_t^{(n)}$.[3] The set $\{(x_t^{(n)}, y_t)\}_{t=1..T}$ forms the training set. In a scoring approach, we train a function f with $f(x_t^{(n)}) \approx y^{(n)}$. In a ranking approach, we train f such that higher-ranked outputs have higher function values. In the example above, we would want: $f(x_t^{(n=1)}) > f(x_t^{(n=3)}) > f(x_t^{(n=2)})$. Once f is trained, it can be applied to rank any new data: this is done by extracting features from references/outputs and sorting by function values.

3 Implementation

3.1 Sentence-level scoring and ranking

We now describe the particular scoring and ranking implementations we examined and submitted to the WMT2008 Shared Evaluation task. In the scoring approach, f is trained using RegressionSVM (Drucker and others, 1996); in the ranking approach, we examined RankSVM (Joachims, 2002) and RankBoost (Freund et al., 2003). We used only linear kernels for RegressionSVM and RankSVM, while allowed RankBoost to produce non-linear f based on a feature thresholds.

[1] Our ranking approach is similar to Ye et. al. (2007), who was the first to advocate MT evaluation as a ranking problem. Here we focus on comparing ranking vs. scoring approaches, which was not done in previous work.

[2] Here we assume single reference for ease of notation; this can be easily extended for multiple reference

[3] Only M_t (not N) features vectors are extracted in practice.

ID	Description
1-4	log of ngram precision, n=1..4
5	ratio of hypothesis and reference length
6-9	ngram precision, n=1..4
10-11	hypothesis and reference length
12	BLEU
13	Smooth BLEU
14-20	Intra-set features for ID 5-9, 12,13

Table 1: Feature set: Features 1-5 can be combined (with uniform weights) to form the log(BLEU) score. Features 6-11 are redundant statistics, but scaled differently. Feature 12 is sentence-level BLEU; Feature 13 is a modified version with add-1 count to each ngram precision (this avoids prevalent zeros). Features 14-20 are only available in the ranking approach; they are derived by comparing different outputs within the same set to be ranked.

The complete feature set is shown in Table 1. We restricted our feature set to traditional BLEU statistics since our experimental goal is to directly compare regression, ranking, and BLEU. Features 14-20 are the only novel features proposed here. We wanted to examine features that are enabled by a ranking approach, but not possible for a scoring approach. We thus introduce "intra-set features", which are statistics computed by observing the entire set of existing features $\{x_t^{(n)}\}_{n=1..M_t}$.

For instance: We define Feature 14 by looking at the relative 1-gram precision (Feature 1) in the set of M_t outputs. Feature 14 is set to value 1 for the output which has the best 1-gram precision, and value 0 otherwise. Similarly, Feature 15 is a binary variable that is 1 for the output with the best 2-gram precision, and 0 for all others. The advantage of intra-set features is calibration. e.g. If the outputs for $r_{t=1}$ all have relatively high BLEU compared to those of $r_{t=2}$, the basic BLEU features will vary widely across the two sets, making it more difficult to fit a ranking function. On the other hand, intra-set features are of the same scale ($[0, 1]$ in this case) across the two sets and therefore induce better margins.

While we have only explored one particular instantiation of intra-set features, many other definitions are imaginable. Novel intra-set features is a promising research direction; experiments indicate that they are most important in helping ranking outperform regression.

3.2 Corpus-level ranking

Sentence-level evaluation generates a ranking for each source sentence. How does one produce an overall corpus-level ranking based on a set of sentence-level rankings? This is known as the "consensus ranking" or "rank aggregation" problem, which can be NP-hard under certain formulations (Meilă et al., 2007). We use the FV heuristic (Fligner and Verducci, 1988), which estimates the empirical probability P_{ij} that system i ranks above system j from sentence-level rankings (i.e. P_{ij} = number of sentences where i ranks better than j, divided by total number of sentences). The corpus-level ranking of system i is then defined as $\sum_{j'} P_{ij'}$.

4 Experiments

For experiments, we split the provided development data into train, dev, and test sets (see Table 2). The data split is randomized at the level of different evaluation tracks (e.g. en-es.test, de-en.test are different tracks) in order to ensure that dev/test are sufficiently novel with respect to the training data. This is important since machine learning approaches have the risk of overfitting and spreading data from the same track to both train and test could lead to overoptimistic results.

	Train	Dev	Test
# tracks	8	3	3
# sets	1504 (63%)	514 (21%)	390 (16%)
# sent	6528 (58%)	2636 (23%)	2079 (19%)

Table 2: Data characteristics: the training data contains 8 tracks, which contained 6528 sentence evaluations or 1504 sets of human rankings ($T = 1504$).

In the first experiment, we compared Regression SVM and Rank SVM (both used Features 1-12) by training on varying amounts of training data. The sentence-level rankings produced by each are compared to human judgments using the Spearman rank correlation coefficient (see Figure 1).

In the second experiment, we compared all ranking and scoring methods discussed thus far. The full training set is used; the dev set is used to tune the cost parameter for the SVMs and number of iterations for RankBoost, which is then applied without modification to the test set. Table 3 shows the aver-

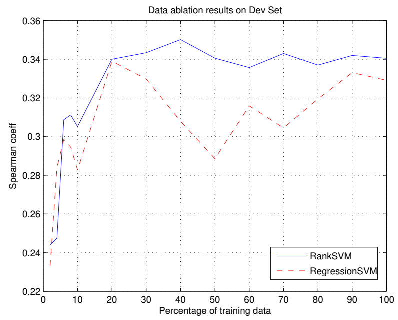

Figure 1: Ranking slightly outperforms Regression for various amounts of training data. Regression results appear to be less stable, with a rise/fall in average Spearman coefficent around 20%, possibly because linear regression functions become harder to fit with more data.

	Feature	Dev	Test
BLEU	1-5	.14	.05
Smoothed BLEU	1-5	.19	.24
Regression SVM	1-12	.33	.24
RankSVM	1-12	.34	.25
RankBoost	1-12	.29	.22
RankSVM	1-20	**.52**	**.42**
RankBoost	1-20	.51	.38

Table 3: Average Spearman coefficients on Dev/Test. The intra-set features gave the most significant gains (e.g. .42 on test of RankSVM). Refer to Table 1 to see what features are used in each row. The SVM/RankBoost results for features 1-12 and 1-5 are similar; only those of 1-12 are reported.

age Spearman coefficient for different methods and different feature sets. There are several interesting observations:

1. BLEU performs poorly, but SmoothedBLEU is almost as good as the machine learning methods that use same set of basic BLEU features.
2. Rank SVM slightly outperforms RankBoost.
3. Regression SVM and Rank SVM gave similar results under the same feature set. However, Rank SVM gave significant improvements when intra-set features are incorporated.

The last observation is particularly important: it shows that the training criteria differences between the ranking and regression is actually not critical. Ranking can outperform regression, but only when ranking-specific features are considered. Without intra-set features, ranking methods may be suffering the same calibration problems as regression.

References

J. Albrecht and R. Hwa. 2007. A re-examination of machine learning approaches for sentence-level MT evaluation. In *ACL*.

S. Banerjee and A. Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *ACL 2005 Wksp on Intrinsic/Extrinsic Evaluation for MT/Summarization*.

J. Blatz et al. 2003. Confidence estimation for machine translation. Technical report, Johns Hopkins University, Natural Language Engineering Workshop.

C. Callison-Burch et al. 2007. (meta-) evaluation of machine translation. In *ACL2007 SMT Workshop*.

S. Corston-Oliver, M. Gamon, and C. Brockett. 2001. A machine learning approach to the automatic evaluation of machine translation. In *ACL*.

H. Drucker et al. 1996. Support vector regression machines. In *NIPS*.

M. Fligner and J. Verducci. 1988. Multistage ranking models. *Journal of American Statistical Assoc.*, 88.

Y. Freund, R. Iyer, R. Schapire, and Y. Singer. 2003. An efficient boosting method for combining preferences. *JMLR*, 4.

T. Joachims. 2002. Optimizing search engines using clickthrough data. In *KDD*.

D. Kauchak and R. Barzilay. 2006. Paraphrasing for automatic evaluation. In *NAACL-HLT*.

A. Kulesza and S. Shieber. 2004. A learning approach to improving sentence-level mt evaluation. In *TMI*.

C.-Y. Lin and F. Och. 2004. Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In *ACL*.

D. Liu and D. Gildea. 2005. Syntactic features for evaluation of machine translation. In *ACL 2005 Wksp on Intrinsic/Extrinsic Evaluation for MT/Summarization*.

M. Meilă, K. Phadnis, A. Patterson, and J. Bilmes. 2007. Consensus ranking under the exponential model. In *Conf. on Uncertainty in Artificial Intelligence (UAI)*.

K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL*.

M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Conf. of Assoc. for Machine Translation in the Americas (AMTA-2006)*.

D. Vilar, G. Leusch, H. Ney, and R. Banchs. 2007. Human evaluation of machine translation through binary system comparisons. In *ACL2007 SMT Workshop*.

Y. Ye, M. Zhou, and C.-Y. Lin. 2007. Sentence level machine translation evaluation as a ranking problem. In *ACL2007 Wksp on Statistical Machine Translation*.