Mach Translat (2008) 22:1–27
DOI 10.1007/s10590-008-9046-1

Regression for machine translation evaluation at the sentence level

Joshua S. Albrecht · Rebecca Hwa

Received: 9 September 2008 / Accepted: 31 October 2008 / Published online: 25 November 2008

Abstract Machine learning offers a systematic framework for developing metrics that use multiple criteria to assess the quality of machine translation (MT). However, learning introduces additional complexities that may impact on the resulting metric's effectiveness. First, a learned metric is more reliable for translations that are similar to its training examples; this calls into question whether it is as effective in evaluating translations from systems that are not its contemporaries. Second, metrics trained from different sets of training examples may exhibit variations in their evaluations. Third, expensive developmental resources (such as translations that have been evaluated by humans) may be needed as training examples. This paper investigates these concerns in the context of using regression to develop metrics for evaluating machine-translated sentences. We track a learned metric's reliability across a 5 year period to measure the extent to which the learned metric can evaluate sentences produced by other systems. We compare metrics trained under different conditions to measure their variations. Finally, we present an alternative formulation of metric training in which the features are based on comparisons against *pseudo-references* in order to reduce the demand on human produced resources. Our results confirm that regression is a useful approach for developing new metrics for MT evaluation at the sentence level.

Keywords Machine translation · Evaluation metrics · Machine learning

J. S. Albrecht · R. Hwa (✉)
Department of Computer Science, University of Pittsburgh, Pittsburgh, PA 15260, USA
e-mail: hwa@cs.pitt.edu

J. S. Albrecht
e-mail: jsa8@cs.pitt.edu

1 Introduction

The establishment of an appropriate evaluation methodology is central to machine translation (MT) research. In order to explore new approaches, an unbiased metric is needed to quantify the degree of improvement. Ideally, the metric will not only judge the translation system's performance as a whole but will also be able to provide more detailed feedback at a finer granularity (e.g., how well the system performed on each sentence). Furthermore, for the evaluation to have a practical impact on the direction of system development, it must also be both computationally efficient and cost-effective. Recent efforts in metric development have been especially concerned with the desiderata of practical efficiency. Widely used metrics such as Bleu (Papineni et al. 2002) predict translation quality by making deterministic measurements of similarity between MT outputs and human translations. Studies have shown these metrics to correlate with human judgments at the document level, but they are less reliable at the sentence level (Blatz et al. 2003). This suggests that the deterministic metrics do not fully reflect the set of criteria that people use in judging sentential translation quality.

One way to capture more judging criteria is to form a composite metric that incorporates numerous indicators, each of which might focus on a particular aspect of the evaluation task. This formulation of the problem lends itself to a machine learning solution: determine how to represent and combine the criteria into a composite metric by optimizing it for a set of translations whose quality is known. To this end, several learning paradigms have been proposed, including classification (Corston-Oliver et al. 2001; Kulesza and Shieber 2004), regression (Quirk 2004), and ranking (Ye et al. 2007). Of the three, regression most directly corresponds to the evaluation task, and it has been shown to be effective for developing metrics under several different conditions (Lita et al. 2005; Albrecht and Hwa 2007a; Liu and Gildea 2007; Uchimoto et al. 2007). On the other hand, regression learning introduces additional complexities to metric development. In this paper, we address several concerns about applying regression for metric development: representation, generalizability, stability, and scalability.

- Knowledge exploration and interpretation: A multitude of judging criteria have been considered for MT evaluation. They may range in scope from lexical choices to syntactic constraints to discourse coherence. Under a learning approach, the extents to which an input translation meets these criteria are expressed as numerical measurements, or *features*, and a composite metric of these criteria is modeled as a mathematical function parameterized by these features. The learning process is a search for parameter values such that the function's outputs are optimized for the training examples. In addition to the choice of features, the success of learning also depends on the choice of the form of the function and the learning algorithm. We conduct experiments to compare metrics represented as linear and nonlinear functions and trained with different learning algorithms. We perform *model tampering* (Goldberg and Elhadad 2007) to analyze the contributions of different knowledge sources.
- Generalization: Perhaps the most central question about the applicability of machine learning to metric development is whether the resulting metric will have a long shelf-life. Because a learned model tends to give more reliable predictions for new

inputs that are similar to its training examples, a valid concern is that the learned metric will become out of date as MT systems improve and produce translations that are very different from those used as training examples. In our experiments, we track how well a learned metric generalizes over a span of five years.
– Stability: The training process introduces a certain degree of variability to metric development. How much does the diversity in training examples impact on the effectiveness of the learned metric? Is the learned metric more suited to the evaluation of certain kinds of MT systems than others? We conduct experiments in which the diversity and variability of the training examples and of the test sentences is controlled. Our results suggest that while some variability is an inherent part of a learning approach, the resulting metrics are relatively stable and robust against training variations.
– Scalability: Finally, we tackle a challenge faced by regression (and supervised learning in general): learning when appropriate training examples are scarce or unavailable. This is especially problematic when the metric is complex (e.g., an expressive function parameterized by a large feature set). Because regression approximates continuous functions, it is important to ascertain how well the approach scales up, and to find ways to minimize the cost of resources for developing training examples. Since human processing is expensive, previous work has proposed feature representations that do not compare against human-produced references (Gamon et al. 2005; Albrecht and Hwa 2007b). Here, we further investigate the use of *pseudo-references*, which are machine-produced translations, as input features. We find that in combination with a learning framework, the metrics that compare against pseudo-references rival standard metrics that use a human reference.

This paper reports a set of experiments that we have designed to address the above concerns. These efforts augment previous work in terms of the scope of our studies. We track metric performance across a longer period of evaluation data, and we make more detailed comparisons between metrics developed under different training conditions. Our results confirm that machine learning is a useful approach for developing new metrics for MT evaluation at the sentence level.

2 Automatic MT evaluation metrics

Many different taxonomies of desiderata and methods of evaluating MT systems have been proposed (Carbonell et al. 1981; Dorr et al. 1999; Hovy et al. 2002). The focus of this paper is restricted to automatic metrics that evaluate MT outputs according to two factors: their *adequacy* in retaining the meaning of the original source text, and their *fluency* in presenting the material in the target language.

2.1 Deterministic metrics

Most deterministic metrics are defined in terms of different ways of comparing against references. First, similarity can be expressed in terms of string edit distances.

In addition to the well-known word error rate (WER), more sophisticated modifications have been proposed to distinguish actual linguistic differences from superficial differences; these include position-independent error rate (PER) (Tillmann et al. 1997), translation edit distance (Snover et al. 2006), and CDer, which allows for block movements (Leusch et al. 2006). Second, similarity can be expressed in terms of common word sequences. Since the introduction of Bleu (Papineni et al. 2002) and NIST (Doddington 2002), the basic *n*-gram precision idea has been augmented in a number of ways. Metrics in the Rouge family allow for skip *n*-grams (Lin and Och 2004a); Kauchak and Barzilay (2006) and Owczarzak et al. (2006) take paraphrasing into account; metrics such as METEOR (Banerjee and Lavie 2005; Lavie and Agarwal 2007) and GTM (Melamed et al. 2003) calculate both recall and precision;[1] METEOR and SIA (Liu and Gildea 2006) also make use of word class information. Finally, researchers have begun to look for similarities at a deeper structural level. For example, Liu and Gildea (2005) developed the Subtree Metric (STM) over constituent parse trees and the Head-Word Chain Metric (HWCM) over dependency parse trees. Owczarzak et al. (2007) have also proposed a syntax-based evaluation metric using dependencies from a Lexical-Functional Grammar parser.

2.2 Learned metrics

Deterministic metrics tend to focus on specific aspects of the evaluation. Machine learning offers a systematic and unified way to combine them into a single metric. The deterministic metrics (or a set of numerical measurements related to a metric) participate as input features for learning. The exact form of the learned metric depends on the representation of the learning problem and the choice of the learning algorithm. In previous work, most learned metrics fall into one of three major families: binary functions that classify whether the input sentence is human-translated or machine-translated (Corston-Oliver et al. 2001; Kulesza and Shieber 2004); continuous functions that score translation quality of input sentences on an absolute scale (Quirk 2004; Lita et al. 2005; Albrecht and Hwa 2007a; Liu and Gildea 2007; Uchimoto et al. 2007); and ordinal functions that give ranking preference between multiple translations (Ye et al. 2007; Duh 2008). Many different measurements have been used as features in developing these learned metrics. In the next section, we briefly summarize some of the more commonly used features.

2.2.1 Features

There are many ways in which judging criteria can be formulated as features. Typically, they are expressed in terms of comparing the input translation against some reference. Perhaps the most informative type of references, as discussed earlier, is a set of well-formed translations produced by humans. More loosely speaking, a large

[1] The parameters controlling the balance between recall and precision for these two metrics can be tuned. In this sense, METEOR and GTM are also learned metrics. The currently publicly available version of METEOR, for instance, has been tuned to improve correlations with human assessments from past NIST MT evaluations (Lavie and Agarwal 2007).

corpus of the target language might also be considered as a kind of reference because it offers a comparison point for determining whether an input translation resembles a sentence in the target language. The degree of similarity is calculated according to different properties of interest. They may be similarities in surface string patterns as well as deeper linguistic similarities. Thus, we group some commonly used features into four categories according to their reference types and the properties they are trying to capture.

2.2.1.1 String comparisons against reference translations Methods for computing string matches against references are widely used both as stand-alone metrics as well as features in a learned metric. For example, Blanc (Lita et al. 2005) can be seen as a parameterized weighting between Bleu and Rouge; similarly, METEOR (Lavie and Agarwal 2007) is a parameterized weighting between precision, recall, and fragmentation. The feature set for the classifier of Kulesza and Shieber (2004) includes WER, PER, as well as features constructed by decomposing Bleu (i.e., raw n-gram matches normalized by sentence length). For the experiments in this paper, our string-reference feature group consists of the following feature types:

- n-gram matches: the number of n-gram matches the input sentence has against human references, normalized by the maximum number of n-gram matches possible (i.e., sentence length $-n + 1$), where $2 \leq n \leq 5$.
- Precision: the percentage of words in the input sentence that matched against the human references.
- Recall: the percentage of words in a human reference that matched the input sentence. If multiple references were given, the reference that is the most similar to the input sentence is used.
- Fragmentation: a ratio that expresses to what extent the matched words are in consecutive chunks (Lavie and Agarwal 2007).
- WER: Levenshtein's minimum edit distance.
- PER: position-independent edit distance.
- Skip-m-bigram matches: nonconsecutive bigrams with a gap size of m, where $1 \leq m \leq 5$.
- Rouge-L: longest common subsequence.

2.2.1.2 Linguistic comparisons against reference translations In addition to using the same words, intuitively, information about whether the input translation and the reference translation use the same syntactic constructs or the same semantic relations should also indicate the degree of similarity between them. In a recent paper by Giménez and Màrquez (2008), deeper linguistic indicators have been combined (without machine learning) into an evaluation metric. These indicators require additional NLP processing on the input and reference translations so that their hidden structures (e.g., parse trees) can be matched. A challenge is that many NLP applications do not expect machine-translated sentences as inputs, so that they often do not produce correct analyses for them. For the experiments in this paper, we focus on matching syntactic constructs because automatic parse analyses can be more reliably obtained than semantic and discourse analyses on translated sentences. Our `linguistic-reference`

feature group consists of the two feature types based on STM and HWCM proposed by Liu and Gildea (2005):

– Subtree matches: this class of features computes the similarity between the constituent parse structures of the translation and the reference. Instead of tallying the number of matches of a sequence of n words, the number of matching subtrees of a depth of d is tallied. We construct a feature for each subtree depth, $2 \leq d \leq 4$.
– Head-word chain matches: this class of features looks for matches in the dependency tree structures of the sentences. Instead of matching on full subtrees over the dependency tree, a sequence of head–modifier relationships is matched instead. For example, in the sentence *John saw a picture of Mary*, *saw* → *picture* → *of* → *Mary* forms a head-word chain of length 4. We construct a feature for each chain length, c, where $2 \leq c \leq 5$. The feature value is the number of head-word chains of length c in a translation sentence that can be found in the references normalized by the number of possible head-word chains of length c in the translation sentence.

2.2.1.3 String comparisons against general corpora This category of features focuses on evaluating the degree of fluency of the translation. An example of this class of features is simply to use the language model score. This was considered by Gamon et al. (2005), who proposed to develop a metric that does not rely on using human-produced translations as references. To recast this type of fluency judgment as some sort of similarity measurement, we can compare characteristics of the input sentence against large target-language corpora. In this work, we construct the feature group `string-general` to be analogous to those in the `string-reference` set, except that a target-language corpus (for our experiments, we used the English Gigaword corpus)[2] is used as the "reference" instead of human translations. Because the emphasis is on checking for fluency, we increased the window of n-grams to range from 2 to 7.

2.2.1.4 Linguistic comparisons over general corpora Similar to the string features, the `linguistic-general` feature set is constructed as a counterpart to the `linguistic-reference` feature set. Instead of parsed human translations, a target-language corpus is automatically parsed to create a "reference" treebank. Because the emphasis of this feature group is on checking for fluency, we included a few more features in addition to subtree matches and head-word chain matches.

– Verb structures: This class of features loosely matches on verb–argument relationships. For each verb in the dependency tree of the input translation, we check whether any of the dependency trees in the corpus treebank contain the verb with the same modifiers. The feature value for the sentence is the percentage of verbs whose modifier structure was found. In addition to exact word matching, we also consider variations in which the modifier words are backed off to their part-of-speech tags and to their grammatical role label.
– Noun structures: This class of features is similar to the verb structure features, but the focus is on nouns and their modifiers in the translation sentence.

[2] Available from Linguistic Data Consortium, Philadelphia PA, catalog number LDC2003T05.

– Head-modifiers: This class of features is another variation in which the head word is backed off to its part-of-speech tag, but the modifiers remain as words. The feature value is the percentage of head nodes whose modifier structures have been found in the target-language corpus treebank.

3 Regression for MT evaluation

An MT evaluation metric can be naturally seen as an ordinal function. Regression provides a straightforward way to learn this type of metric. In regression learning, we are interested in approximating a function f that maps a d-dimensional input feature vector, $\mathbf{x} = x_1, \ldots, x_d$, to a continuous real value, y, such that the error over a set of m training examples, $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)\}$, is minimized according to a loss function. A commonly used loss function is the quadratic error function, as in (1).

$$\frac{1}{2}\sum_{i=1}^{m}(y_i - f(\mathbf{x}_i))^2 \tag{1}$$

In the context of MT evaluation, y is the "true" quantitative measure of translation quality for an input sentence.[3] The function f represents a mathematical model of human judgments of translations; an input sentence is represented as a feature vector, $\mathbf{x}$, which encodes a collection of criteria for judging the input sentence, such as those described in Sect. 2.2.1, that are relevant for computing y.

One concern raised about applying regression to MT evaluation is its scalability, since a large set of training examples may be required to fit an arbitrary continuous function (Kulesza and Shieber 2004). We argue that regression can be made more tractable. Because we have some knowledge about the domain of the problem, we can make reasonable assumptions to place constraints on the form of the target function. For instance, if the input features are binary indicators that represent a collection of independent desiderata for a good translation, the metric can take the form of a linear combination of all the feature measurements weighted by a vector of parameters $\mathbf{w}$ (with b as the bias), as in (2).

$$f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b \tag{2}$$

The advantage of modeling the metric as a linear function is that the number of training examples needed is relatively small. It is proportional to the dimension of the feature vector $(d + 1)$.

Nonlinear interactions between the input measurements can be modeled by mapping them to a more complex feature space by some nonlinear transformation function $\Phi(\mathbf{x})$. A simple example of a $\Phi(\mathbf{x})$ is one that consists of pairwise products of individual

[3] Perhaps even more so than grammaticality judgments, there is variability in people's judgments of translation quality. However, like grammaticality judgments, people do share some similarities in their judgments at a coarse-grained level. Ideally, what we refer to as the true value of translational quality should reflect the consensus judgments of all people.

features ($x_i \times x_j$, where $1 \leq i, j \leq d$), but for many problems, the exact form of the Φ transformation function does not have to be explicitly specified. For example, in a feed-forward neural network, the hidden layers can be seen as a representation of $\Phi(\mathbf{x})$. Once the input feature vector has been projected up to this transformed feature space, the metric can be learned once again as a linear function, as in (3).

$$f(\mathbf{x}) = \mathbf{w}' \cdot \Phi(\mathbf{x}) + b' \tag{3}$$

Note that the new parameter vector, $\mathbf{w}'$, has the same dimensionality as the transformed feature vector, $\Phi(\mathbf{x})$, which is usually much larger than the original d.

An alternative representation of the function f is to express it in terms of a linear combination of comparisons between an input instance, $\mathbf{z}$, and the training examples ($\mathbf{x}_1 \ldots \mathbf{x}_m$), as in (4),

$$f(\mathbf{z}) = \sum_{i=1}^{m} \alpha_i y_i \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{z}) + b \tag{4}$$

where α_i are the parameters associated with each training example and b is the bias. Moreover, for many feature transformation functions $\Phi(\mathbf{x})$, there exists some *kernel function*, K, that equals the inner product of the feature vectors of the input and the training examples, as in (5),

$$K(\mathbf{x}, \mathbf{z}) = \Phi(\mathbf{x}) \cdot \Phi(\mathbf{z}) \tag{5}$$

but $K(\mathbf{x}, \mathbf{z})$ can be computed more efficiently as a direct function of the input vectors $\mathbf{x}$ and $\mathbf{z}$.

Support vector machines (SVMs) leverage this form of representation and efficient kernel function computations to learn nonlinear functions (Cortes and Vapnik 1995). SVMs are most commonly trained for binary classification, in which the learned function f is a dividing hyperplane that provides the largest margin between the positive and negative training examples. For the experiments in this paper, we use SVMs for regression so that f is the target function that outputs a scalar prediction of translation quality. During training, support vector regression aims to minimize an ϵ-insensitive error function as its loss function (cf. Eq. 1). An ϵ-insensitive error function allows for errors within the margin of ϵ, a small positive value, to be considered as having zero error (cf. Bishop 2006, pp. 339–344). We compare trained metrics based on the following three kernel functions (cf. Hastie et al. 2001) that map to feature transformations of increasing expressivity:

– linear kernel: this is the identity case where the input features are not transformed (6).

$$K(\mathbf{x}, \mathbf{z}) = \mathbf{x} \cdot \mathbf{z} \tag{6}$$

– p-degree polynomial kernel: the input features are transformed such that $\mathbf{\Phi}(\mathbf{x})$ contains all p-way products of the original input features (7),

$$K(\mathbf{x}, \mathbf{z}) = (1 + \mathbf{x} \cdot \mathbf{z})^p \tag{7}$$

– Gaussian kernel: the input features are mapped to a transformed space of infinite dimensionality (8),

$$K(\mathbf{x}, \mathbf{z}) = \exp(-|\mathbf{x} - \mathbf{z}|^2/g) \tag{8}$$

where g controls the variance of the Gaussian.

3.1 Feature analysis

One of the motivations for taking a machine learning approach to metric development is to gain insights into the relative importance of different criteria and how they come together to form a metric. Therefore, feature analysis is an important part of the metric development process.

One way to perform an analysis is to examine directly the weight parameters associated with the features. As a rule of thumb, a large value placed on a weight should indicate that the corresponding feature is important; however, this is not always the case because not all features share the same value ranges. In other words, a weight parameter might converge on a large value to compensate for a feature that tends to take on small values. Moreover, for nonlinear functions, it is less intuitive to determine the relative importance of features from the parameter values. For instance, the values of the parameters α in Eq. (4) indicate which of the associated training examples are important, but they do not explicitly point out which features contributed the most to the weighting of these examples. Therefore, an analysis of the weight values would require additional conversions and normalizations. Finally, the weight values reveal their relative importance to the learned model, but they are not directly linked to the model's performance.

A simple but effective alternative is model tampering (Goldberg and Elhadad 2007). With this approach, a model (i.e. the function) is first trained in the usual manner, and we assess how well the model fits the training data. Next, we modify the model to prevent the chosen features from having any effect on the prediction of test instances. In models such as perceptrons (see Eq. 2), this can be achieved by zeroing out the weight parameters associated with chosen features; for SVMs, since all the training examples are embedded in the model (see Eq. 4), this can be achieved by directly zeroing out the values of the chosen features for all training examples. Comparing the performance of the tampered model with the original, we obtain a quantitative estimate regarding the impact of the tampered features on performance.

Model tampering differs from feature selection because the model is modified *after* training rather than *before*. By tampering with the features after training, we can observe the importance of different features on the same trained model. A second

benefit is the simplicity of the approach. Unlike standard feature selection strategies, which may require multiple iterations of retraining with different feature subsets, the post-training tampering is orders of magnitude faster.

3.2 Learning with pseudo-references

Many of the measurements described in Sect. 2.2.1 are comparisons between the input translation and human reference translations. Each numeric measurement can be thought of as a distance, describing how far away the input is from a known good translation according to some criterion. Since there are usually multiple acceptable translations, automatic deterministic metrics such as those described in Sect. 2.1 (and thus, the features based on them) are more reliable when they compare the input against more than one reference. This is problematic because creating references requires bilingual humans, not just for the training examples, but also for evaluating new inputs with the trained metrics.

For a formally organized event, such as the annual MT evaluation program sponsored by National Institute of Standard and Technology (NIST), it may be worthwhile recruiting multiple human translators to translate a few hundred sentences for evaluation references. However, there are situations in which multiple human references are not practically available (e.g., the source may be of a large quantity, and no human translation exists). One such instance is translation quality assurance, in which one wishes to identify poor outputs in a large body of machine-translated text automatically for humans to postedit. Another instance is in day-to-day MT research and development, where new test sets with multiple references are also hard to come by. One could work with previous datasets from events such as the NIST MT evaluations, but there is a danger of overfitting. One also could extract a single reference from parallel corpora, although it is known that automatic metrics are more reliable when comparing against multiple references.

To reduce the demands on humans for generating references, we consider whether sentences produced by MT systems might stand in as a kind of a "pseudo-reference" for human references. For example, one might use commercial off-the-shelf systems, some of which are freely available over the web. For less commonly used languages, one might use open-source research systems (Al-Onaizan et al. 1999; Burbank et al. 2005). Of course, pseudo-references are not perfect themselves; thus, even if an input translation were identical to a pseudo-reference, it might not be a good translation. The key shift in perspective is that we view the reference not as an ideal to strive for but as a benchmark to compare against. We hypothesize that comparing MT outputs against imperfect translations from MT systems that use different approaches may result in a more nuanced discrimination of quality.

As a toy example, consider a one-dimensional line segment. A distance from the end-point uniquely determines the position of a point. When the reference location is anywhere else on the line segment, a relative distance to the reference does not uniquely specify a location on the line segment. However, the position of a point can be uniquely determined if we are given its relative distances to two reference locations.

The problem space for MT evaluation, though more complex, is not dissimilar to the toy scenario. There are two main differences. First, we do not know the distance of translation quality between two sentences; this is what we are trying to learn. The distances we have at our disposal are all heuristic approximations to the true translational distance (i.e., the features). Second, unlike human references, whose quality is assumed to be maximally high, the qualities of the pseudo-reference sentences are not known. In fact, we may not even have a precise estimation of the quality of the MT systems that produced the pseudo-reference sentences.

Both issues of combining features and deciding how much to trust each pseudo-reference system can be addressed through machine learning. As with normal regression learning, the training data is a set of MT outputs whose qualities have already been judged by humans, but the feature vector of each example is made up of measurements between the pseudo-references and the assessed MT output. If many of a reference system's outputs are similar to those MT outputs that received low assessments, we can conclude that this reference system may not be a high quality system. Furthermore, if a new translation is found to be similar to this reference system's output, it is more likely for the new translation also to be bad.

4 Experimental methodology

To gain a better understanding of the strengths and weaknesses of using regression to develop metrics for MT evaluation at the sentence level, we designed five experiments to compare the performance of the metrics that are trained and tested under a variety of controlled conditions.

- We compare different algorithms for regression learning. To consider functions of different expressivity, we trained an SVM with a linear kernel, with a polynomial kernel, and with a Gaussian kernel. The implementation used in our experiments is SVM-Light (Joachims 1999). Additionally, we tried two other nonlinear function approximation methods: neural networks and regression trees, using the WEKA implementation for both (Witten and Frank 2005).
- In order to develop better future MT systems, it is helpful to know which features contribute more heavily to the metric function. We perform feature analysis on different subgroups to measure their contribution to the learned metrics.
- We measure the learned metric's ability to generalize to novel inputs. The training examples for the learned metric are outputs of Chinese–English MT systems from a particular year; we test the metric on sentences produced by Chinese–English MT systems from different years as well as sentences produced by Arabic–English MT systems.
- Unlike deterministic metrics, the outputs of a learned metric may show some variation depending on the training examples. To understand the degree of the variations, we compare metrics that were developed under different training conditions, controlling for the diversity and quality of the training examples. We measure the extent to which the resulting metrics can reliably provide the same levels of predictions for a large heterogeneous test set. We also examine the performance of the metrics on sentences from individual MT systems.

– To investigate whether learning with features extracted with respect to pseudo-references instead of human references is a viable approach, we compare the learned metrics of the two representations in terms of stability and generalization.

This section describes the common experimental data and methodologies used by all the studies.

4.1 Data

Our experimental datasets are taken from five years of NIST MT evaluations (2002–2006). Each year's dataset consists of a set of source sentences, four human reference translations for each source sentence, translations from participating MT systems, and human assessments of those sentences. The texts used in these datasets are from different news sources. The 2002–2005 datasets contain newspaper articles; the 2006 dataset contains both newspaper articles and online newsgroup articles. The generalizability and the stability of the learned metrics with respect to genre variations is thus beyond the scope of the experiments reported in this paper.

Table 1 summarizes some statistics of the datasets. All sentences from the participating MT systems have been evaluated by at least two human judges. For the datasets from years 2002–2005, each judge assigned a separate score for fluency and for adequacy; in 2006, the judges assigned only a single composite score. Each score is given as an integer from 1 to 5. Because each human judge has a somewhat different standard, the collected scores may be from differently biased distributions. To reduce the impact of these variations, we normalize the human judges' scores following the process described by Blatz et al. (2003), who found score normalization to increase the correlations between human judges.

To perform this normalization, each score (s) by a given human judge (J) is converted into a quantile (x), which represents the normalized score. The quantile can be computed as $P(S < s|J)$, the probability that judge J would assign a score less than s. This would simply be the number of scores by J that were less than s divided by the total number of scores by J. An alternative would be to consider the scores that

Table 1 Some statistics about the datasets used for the experiments

The number of sentences assessed is the product of the number of source sentences, the number of MT systems and the number of human judges per sentence

Year	Source language	No. of source sentences	No. of MT systems	No. of sentences assessed
2002	Chinese	879	3	5,787
2003	Arabic	663	2	2,652
2003	Chinese	919	6	11,028
2004	Arabic	347	10	6,940
2004	Chinese	447	10	8,940
2005	Arabic	266	7	3,724
2005	Chinese	272	7	3,808
2006	Chinese	300	5	3,000

were less than or equal to s. The final normalized score, $\hat{x}$, can be seen as a mix of these two approaches ($P(S < s|J)$ and $P(S \leq s|J)$) (9).

$$\hat{x} = \frac{\left(\sum_{i=1}^{s-1} n(i)\right) + n(s)/2}{\sum_{i=1}^{\text{MaxScore}} n(i)} \tag{9}$$

where $n(i)$ is the number of times that judge J gave a score of i to the sentences in the dataset.

For the years in which both fluency and adequacy were judged, the scores are first normalized separately. We compute a judge's overall score for a sentence by normalizing the sum of the raw fluency and adequacy scores (i.e., MaxScore = 10), whereas in the 2006 dataset, the single score is directly normalized. For evaluating the learned metric, we consider the "gold standard" score for a translation output to be the average of all judges' overall scores for that sentence.

For all but one experiment, we use the 2003 Chinese–English set as training data because it is the earliest year that had a large number of human-assessed sentences. The exception is in the stability study for learning from pseudo-references; in that experiment, we worked with the 2004 Chinese–English set because it has the most participating systems. In all experiments, we reserve one fifth of the training data as held-out data for parameter tuning for the learning algorithms (e.g., the slack variable and the width of the Gaussian for the SVM).

4.2 The evaluation of evaluation metrics

The success of a metric is determined by how closely its scores for the translation qualities of the input sentences agree with those given by human judges for the same sentences. We compute the Spearman rank-correlation coefficient between the metrics' scores and the averaged human assessments on a set of test sentences. We use Spearman instead of Pearson because it is a distribution-free test. The coefficient, denoted as ρ, is a real number ranging from -1 (indicating perfect negative correlations) to $+1$ (indicating perfect positive correlations). A good metric, therefore, should correlate positively with human judgment.

It is not straightforward, however, to decide whether one metric is better than another based on the correlation coefficients directly. That is, we cannot simply conclude that Metric_A with $\rho_A = 0.65$ is necessarily better than Metric_B with $\rho_B = 0.62$. To gather more reliable statistics for making comparisons between metrics, we need multiple test trials to perform hypothesis testing to determine whether the difference between the metrics' correlation coefficients is statistically significant. One commonly used strategy is to generate multiple test trials out of one test set via bootstrap resampling (Koehn 2004); however, it has been criticized for being overly optimistic due to its assumptions about the sample distribution (Riezler and Maxwell 2005). Riezler and Maxwell proposed to use the *approximate randomized test* as an alternative. The idea behind this test is that if the null hypothesis (that the two metrics have the same evaluative capabilities) were true, then random swaps of met-

ric scores on some test instances would have no adverse effect on the delta difference between the correlation coefficients. Let p be the percentage of trials in which no adverse effect was detected. The null hypothesis is rejected if p is less than or equal to the desired rejection threshold. We have implemented both and empirically observed the approximate randomized test to be more conservative than the bootstrapping resampling method; thus in the following experiments we use the former to compute statistical significance. For statistical tests in our experiments, 500 trials are conducted, and we consider two metrics to be qualitatively different if $p \leq 0.05$.

As baseline comparisons, we report the correlations of three automatic metrics: Bleu, which is precision-centric; METEOR,[4] which incorporates recall and stemming; and HWCM, which uses syntax. Bleu is smoothed to be more appropriate for sentence-level evaluation (Lin and Och 2004b), and the bigram versions of Bleu and HWCM are reported because they have higher correlations than when longer n-grams are included. This phenomenon was previously observed by Liu and Gildea (2005).

5 Results and discussions

Five sets of experiments were conducted to investigate the choice of learning algorithms, the role of the features, the learned metric's ability to evaluate translations from future systems, the impact of variations in the distribution of training examples on the metric's effectiveness in evaluating different test systems, and whether training with pseudo-references is a viable option for metric development.

5.1 The choice of learning algorithm

In this first experiment, we compare the effects of training with different function forms and learning algorithms. Although the learning models differ, they share the same input feature vector, which consists of all the measurements presented in Sect. 2.2.1.

The learned metrics are all trained and tested in the same manner. We performed five-fold cross validation on the NIST 2003 Chinese–English dataset. The sentences are randomly grouped into five subsets. For each fold, four subsets are used for training while the fifth is reserved for testing. Overall, each sentence in the dataset is evaluated once by a metric that has not seen it during training.[5] The correlation coefficient is then computed comparing the trained metrics' outputs against human assessments for the full dataset.

The results are summarized in Table 2. As points of comparison, the results of the baseline deterministic metrics are shown in the first column. Each of the remaining columns presents a family of learning algorithms; SVM metrics of different kernels are in the second column; metrics modeled by additional learning algorithms such as a

[4] In this paper, we used the earlier METEOR proposed by Banerjee and Lavie (2005) rather than the latest downloadable (version 0.6) because its tuning data overlaps with our test dataset.

[5] Note, however, that the metric has seen other instances from the same MT system that produced the test instance. We evaluate the metric's ability to evaluate outputs from new MT systems in later experiments.

Table 2 A comparison of metrics trained with different learning algorithms

Baselines		SVM Kernels		Other algorithms	
Bleu	0.465	Linear	0.501	Neural net (two layers)	0.501
METEOR	0.480	Poly ($p = 3$)	0.501	Regression tree	0.461
HWCM	0.412	Gaussian	0.502		

We performed five-fold cross validation on the NIST 2003 Chinese dataset and computed the Spearman correlation coefficient between each metric's scores and the human assessment scores for the dataset. With the exception of learning with regression tree, the learned metrics performed significantly better than the baselines, but they are not distinguishable from each other

Table 3 A comparison of the impact of different input feature groups on the learned metric

Feature group	Remove this group	Use only this group
All the features	–	0.512
`String-references`	0.428	*0.499*
`Linguistics-references`	*0.509*	0.415
`String-general`	*0.507*	0.064
`Linguistic-general`	*0.508*	0.126

For each feature group, we observe the change in the Spearman correlation coefficient when all its features are disabled, and when only its features are allowed. The values that are not significantly different from the original correlation coefficient (0.512) are shown in italics

two-layer feed-forward neural network and regression trees are presented in the third column.

We observed that while nearly all the learned metrics correlated better with human assessments than the baseline metrics, the choices of the form of the function and the learning algorithm do not have a perceivable effect on the resulting metric. This may be because many features are individual indicators of translation quality, and can be combined by a linear function such that the more expressive functions do not offer additional benefits. To reduce the possibility of overfitting to the training examples, for the rest of the experiments in this paper, we train the regression metrics using SVM with a linear kernel.

5.2 Feature analysis

In Sect. 2.2.1, we presented an overview of four groups of features that have been commonly used in previous work on evaluation metrics. In this experiment, we aim to quantify the contribution of these features.

We first train the metric with the entire NIST 2003 Chinese dataset. We then estimate an upper limit for the learned metric's performance by testing it on its training data. Next we modify the model to demonstrate the impact of the chosen features. We retest the modified model and recompute the correlation coefficients. This study is conducted with respect to the training dataset rather than a new test set because its purpose is to determine which set of features have been deemed

important by the training process, rather than which tampered model generalizes better.

Table 3 presents the changes in correlation coefficients. The correlation coefficient between the original metric's prediction and the human assessments is 0.512. If the tampered features were important, the modified metric would become less correlated with human assessments. For each of the four main feature groups, we modified the model in two ways: removing the group by zeroing out its feature values, and the complement, keeping only that group by zeroing out all the other features.

Of the four groups of features, the learned metric relies the most on string comparisons against references. Using only features of this group, the metric's correlation is slightly lowered (to 0.499); without the features of this group, the metric's correlation dropped significantly. When the other feature groups are excluded, the metric's correlation with human assessment was not significantly worsened. This suggests that these features served a more auxiliary purpose in the learned metric. However, it does not necessarily mean that these features are not informative. When all the features from `string-reference` are disabled, the metric still has a correlation coefficient of 0.428.[6] The results of this experiment suggest that the full feature set contains some informational redundancies, but taken together, the features contribute to a more informed metric than the baseline metrics.

5.3 Generalization

The results from the cross-validation experiments showed that regression can train competitive metrics to evaluate new translations from the MT systems that generated the sentences used as training examples. For a metric to be effective, however, it should provide reliable assessments to outputs from different systems. In this experiment, we measure the learned metric's ability to generalize to new data.

A metric is developed using the NIST 2003 Chinese–English dataset as training examples. We then apply it to evaluate sentences produced by two different types of MT systems: Arabic–English systems and Chinese–English systems from different years (all from NIST evaluations). If the metric were fitted too closely to the training examples, we would expect it quickly to become ineffective at judging the newer, more advanced MT systems.

Table 4 presents a summary of the results.[7] We computed the correlation coefficients between the metric's assessments and the human assessments for sentences from the same NIST dataset. We also computed the corresponding correlation coef-

[6] Note that this value is a somewhat pessimistic estimation of the usefulness of the enabled features because the metric has been optimized for the full feature set during training. The correlation coefficient should be somewhat higher if the metric were optimized with the reduced feature set during training, in the manner of feature selection.

[7] This experiment is a modified and expanded version of our earlier work (Albrecht and Hwa 2007a). Previously published results roughly correspond to the first three lines of this table. In that earlier study, three references were used for feature calculation (because during training, the fourth reference was used as a positive example for the classification metric); here, all four references are used.

Table 4 A comparison between baseline metrics and a regression-trained metric on sentences produced by different MT systems over multiple years

Test set	Bleu	METEOR	HWCM	Regr (2003 Chn)
2002 Chn	0.283	0.298	0.272	**0.337**
2003 Ara	0.460	0.454	0.457	**0.505**
2004 Chn	0.593	0.568	0.551	**0.616**
2004 Ara	0.577	0.580	0.521	**0.611**
2005 Chn	0.489	0.509	0.457	**0.565**
2005 Ara	0.427	0.427	0.376	**0.459**
2006 Chn	0.475	**0.525**	0.337	*0.506*

The regression metric is trained on the NIST 2003 Chinese–English dataset. The table reports the Spearman correlation coefficients between a metric's scores and the human assessment scores for the test sentences. The highest correlation coefficient for each test set is highlighted in bold font, and the coefficients that are not found to be significantly worse are italicized

Table 5 The six systems that were evaluated by human judges in the NIST 2003 Chinese–English MT evaluation and the averaged sentence-level score that each system received

SysID	Average human assessment score
03-C1	0.636
03-C2	0.562
03-C3	0.543
03-C4	0.498
03-C5	0.446
03-C6	0.315

ficients for the baseline metrics. The results show that the regression-trained metric maintains a higher correlation coefficient than the baseline metrics for nearly all years. This suggests that the learned metric generalizes well and is broadly applicable to a wide range of MT systems.

As discussed earlier, the learned metric can be seen as a composite of the deterministic metrics because it draws from much of the same information sources. The consistent lead over the baseline metrics indicates that regression training is successful in combining these indicators into one metric. As MT research continues to advance, the baseline metrics may become less effective, and therefore the evaluative power of this particular learned metric may also wane. In that case, regression training can be used to develop a new metric from a new set of relevant features.

5.4 Stability

To evaluate the stability of the learning approach for metric development, we compare several learned metrics that were developed with training examples from MT systems of different quality. In this study, the sets of training examples are selected from the NIST 2003 Chinese–English dataset; all metrics are tested on the same test cases:

Table 6 A comparison of metrics that were developed using subsets of the full NIST 2003 Chinese–English training set

	No 03-C1	No 03-C2	No 03-C3	No 03-C4	No 03-C5	No 03-C6
2002 Chn	*0.346*	*0.331*	*0.336*	*0.345*	*0.327*	*0.331*
2003 Ara	0.486	*0.510*	*0.497*	*0.508*	*0.504*	*0.520*
2004 Chn	0.591	**0.624**	*0.613*	*0.616*	0.605	**0.634**
2004 Ara	0.591	*0.611*	0.591	*0.604*	*0.605*	*0.608*
2005 Chn	*0.553*	*0.558*	0.546	*0.563*	0.549	*0.559*
2005 Ara	0.418	*0.457*	*0.452*	*0.448*	0.447	*0.463*
2006 Chn	*0.498*	*0.500*	*0.520*	*0.516*	**0.519**	0.468

Each column presents a metric's correlation coefficients on different test sets. The header of each column specifies the training condition of its metric. For instance, "No 03-C1" means that sentences produced by the MT system *03-C1* were excluded from the training examples. The coefficients that are significantly higher than the similarly computed coefficients of the regression metric in Table 4 are shown in bold font; the ones that are not significantly worse are shown in italics

Table 7 The ten systems that were evaluated by human judges in the NIST 2004 Chinese–English MT evaluation and the averaged sentence-level score that each system received

SysID	Average human assessment score
04-C1	0.661
04-C2	0.626
04-C3	0.586
04-C4	0.578
04-C5	0.537
04-C6	0.530
04-C7	0.530
04-C8	0.375
04-C9	0.332
04-C10	0.243

the NIST 2004 Chinese–English dataset. We prepared six training sets; in each set, one of the six assessed MT systems in the NIST 2003 dataset is withheld. Table 5 lists the averaged human assessment score of each system (we assigned each system an ID based on the rank of its human assessment scores). As the table shows, these systems spanned a range of performance levels. Thus, when *03-C1* is left out of the training set, the metric is tuned on MT sentences that received lower human assessments; in contrast, when *03-C6* is left out, the training set is biased towards good MT sentences. The goal of this experiment is to see how much the bias might impact on the reliability of the metric.

Table 6 presents the results. This table can be seen as an extension of Table 4 from the previous generalization experiment. Although the six metrics reported in this table are trained on 83% of the examples used to train the regression metric in Table 4, the smaller training size in and of itself does not seem to have a major negative impact. Of the six metrics, we observe that the correlation coefficients of *No 03-C1* are typically

lower than those from the regression metric of Table 4 and the other metrics. The results confirm our intuition that it is important to have sentences with a wide range of assessment quality as training examples. Moreover, the results suggest that it is more helpful to have sentences from higher quality MT systems as training examples. That is, having seen extra training examples from a bad system is not as harmful as having not seen training examples from a good system. This is also consistent with our intuition. Since there are many ways to create bad translations, seeing a particular type of bad translation from one system may not be very informative. In contrast, the neighborhood of good translations is much smaller, and is what all the systems are aiming for; thus, assessments of sentences from a good system can be much more informative.

Another concern for the learned metric's stability is whether the learned metric might be systematically biased in favor of (or against) certain types of test MT systems. We further explore the relationship between variations in training data and the metric's ability to evaluate different MT systems. Specifically, we examine the fourth column of Table 6 in more detail. In the previous experiment, we evaluated the learned metrics against the entirety of the NIST 2004 Chinese–English dataset; here we compute the metrics' correlation coefficients for the ten MT systems individually. We chose to focus on the NIST 2004 Chinese–English dataset because it has the largest number of human-evaluated MT systems. The systems' averaged human assessment scores are shown in Table 7.

Table 8 summarizes the results.[8] Each column presents the correlation coefficients for the ten test systems using one metric. We compare the three baseline metrics and three variations of the regression metrics: trained on the best five Chinese–English MT systems from 2003 (excluding *03-C6*); trained on the worst five systems (excluding *03-C1*); and trained on all six systems. For each test system (per row), we display the metric that has the highest correlation coefficient in bold font, and we italicize all the other metrics' correlation coefficients that were not found to be significantly worse according to the statistical test. Note that in this experiment, the correlation coefficients are computed from a smaller dataset (447 instances). Because the approximate randomized test is relatively cautious, a larger difference between two metrics' correlation coefficients has to be established than in previous experiments for the difference to be deemed significant.

The results of this experiment reinforce some of the observations made earlier. We find that the regression metrics typically have higher correlation coefficients than the baselines. Of the three regression metrics, *No 03-C1* often has a slightly lower coefficient than the other two. It was also not as effective at evaluating the higher quality MT systems (*04-C1* and *04-C2*), perhaps because it was not trained on as many high quality translations. In contrast, *No 03-C6* tends to have higher correlation coefficients, even for lower quality MT systems. However, the differences between the regression metrics are generally not large enough for the statistical test to conclude a significant difference. This suggests that human assessments of good translations

[8] This experiment is an update of our earlier work (Albrecht and Hwa 2007a). As mentioned in footnote 6, the numerical differences are due to the slight change in the experimental setup.

Table 8 A comparison of metrics by computing their correlation coefficients for each test system

Test system	Bleu	METEOR	HWCM	No 03-C6 (top 5 sys)	No 03-C1 (bottom 5 sys)	03-all (all 6 sys)
04-C1	0.464	0.464	0.445	**0.537**	0.465	*0.509*
04-C2	0.370	0.343	0.363	**0.449**	0.371	0.399
04-C3	0.365	0.400	0.360	**0.468**	*0.417*	*0.442*
04-C4	*0.422*	0.421	0.378	**0.477**	*0.435*	*0.462*
04-C5	0.363	*0.455*	0.335	**0.484**	*0.468*	*0.478*
04-C6	*0.403*	0.365	*0.388*	*0.402*	*0.408*	**0.417**
04-C7	*0.400*	*0.412*	*0.380*	**0.454**	*0.429*	*0.447*
04-C8	*0.239*	*0.255*	0.167	**0.311**	*0.289*	0.274
04-C9	0.520	*0.552*	0.537	*0.590*	0.579	**0.602**
04-C10	*0.318*	0.316	*0.377*	**0.389**	*0.348*	*0.371*

The columns specify which metric is used. The rows specify which MT system is under evaluation. For each evaluated MT system (row), the highest coefficient is shown in bold font, and those that are statistically comparable to the highest are shown in italics

may make more helpful training examples but that some variation in training does not have a large impact on the learned metrics.

In terms of the impact of the test systems, the results suggest that all six metrics are biased against the lower-quality MT systems to some extent. The correlation coefficients between the metrics and the human assessments are the lowest for MT systems *04-C8* and *04-C10*. Moreover, the improvements of the regression metrics over the baselines for the lower-quality systems are not as obvious. This is because the baseline metrics' criteria are encoded as input features for the regression metrics. Thus, when all the baseline criteria are unreliable, the regression metric would also make bad predictions. Because judging criteria are more naturally expressed as desiderata for good translations, it may be harder for quantitative assessments of a bad translation to agree.

5.5 Metrics that compare against pseudo-references

Thus far, our experimental results suggest that regression can be an effective training method for developing complex metrics for evaluating sentences from a diverse set of MT systems. In terms of scalability, the amount of resources needed for the training infrastructure is arguably not an exorbitant investment. In the previous experiments, metrics were trained relying only on the NIST 2003 Chinese–English dataset, which consists of 11,028 instances of human assessment scores. The hidden cost, however, is that many of the features require comparisons against multiple reference translations. Since test instances have the same feature representation, it means that multiple references are always necessary (this is true for Bleu and other deterministic metrics as well).

We have argued that in a machine learning setting, metrics can be trained with feature values computed from pseudo-references. To investigate whether the proposed approach is viable, we conduct two experiments. The first examines the role of learn-

ing. We measure the impact of the variation in the quality of the pseudo-reference MT systems on evaluation metrics. In the second experiment, we explore whether the pseudo-reference training approach is effective for metric development in practical situations; we evaluate the learned metrics over diverse test sets in a manner similar to the studies in Sect. 5.3.

For these experiments, we made two modifications to the feature representation to adapt the learning framework from using human references to using pseudo-references. First, we did not include the `linguistic-reference` features. Because the automatically produced parse structures for the MT references may be too unreliable, comparisons of structural similarities between a candidate translation and MT references are unlikely to be strong indicators. Second, in order to allow the model to learn to differentiate between the qualities of multiple pseudo-references, a feature is individually associated with every reference, whereas in the case with human references, a feature value is computed either as an aggregate over all references (e.g., Bleu-style n-gram matches) or as the value of the best matching reference (e.g., recall).

5.5.1 Impact due to the quality of the MT references

In this study we examine the interaction between pseudo-references and machine learning in a controlled setting. Therefore, the pseudo-references, the training and the test instances are all taken from the NIST 2004 Chinese–English dataset because it contains the highest number of participating MT systems and the systems span over a range of performance levels (see Table 7 for a ranking of the systems and their averaged human assessment scores). Specifically, we reserved four systems (*04-C2*, *04-C5*, *04-C6*, and *04-C9*) for the role of pseudo-references. Sentences produced by the remaining six systems are used as evaluative data. We perform five-fold cross validation on the evaluative dataset. Baseline metrics can also use pseudo-references without learning; as points of comparison, we also score all the sentences of the evaluative set with Bleu and METEOR using pseudo-references.

Table 9 presents a comparison of the different metrics' performance (in terms of correlation coefficients) on the six-system evaluative dataset given different reference configurations. The reference configurations are varied in terms of their number, type, and quality. For the case when only one human reference is used, the reference was chosen at random from the 2004 NIST evaluation dataset.[9]

Some trends are as expected; comparing within a metric, having four human references is better than having just one, and having high-quality systems as references is better than having low-quality systems as references. Perhaps more surprising is the trend that metrics do significantly better with four MT references than with one human reference. This is consistent with the common wisdom that a metric would have a great variation in reliability if it is based on one reference. Moreover, when trained

[9] In the NIST datasets, human reference translations were not assessed by human judges. To get a feel for the quality of the references, we compared each reference against the other three with deterministic MT evaluation metrics. We rank this particular translator third, but the quality of all four human references are significantly higher than that of the best MT systems.

Table 9 A comparison of metrics (columns) using different types of references (rows)

Ref type and #	Ref quality	Bleu	METEOR	Regression
4 Human	–	0.627	0.591	**0.672**
1 Human	Human Ref #3	0.531	0.512	0.623
4 Systems	–	0.612	0.583	*0.666*
2 Systems	Best 2 MT Refs	0.601	0.577	0.651
	Mid 2 MT Refs	0.577	0.555	0.644
	Worst 2 MT Refs	0.539	0.508	0.626
1 System	Best MT Ref	0.575	0.560	0.636
	Mid MT Ref (04-C5)	0.531	0.529	0.627
	Worst MT Ref	0.379	0.329	0.575

The full regression-trained metric has the highest correlation coefficient value (shown in boldface) when four human references are used. When four MT system references are used, the coefficient is slightly decreased (shown in italics), but our statistical test could not conclude that the difference is significant ($p = 0.39$)

with regression, the metric that uses four MT references has a correlation coefficient that is only slightly lower than that of the metric trained to use four human references such that the statistical test could not conclude that the metrics are significantly different ($p = 0.39$).

Another observation is that while MT references can be used for standard metrics such as Bleu and METEOR, they may not be appropriate. Because they treat each reference as a gold standard, their reliability suffers more when the MT references are bad. In the regression-trained metric's case, it learns to assign higher weights to the more helpful features. Further, the learned metrics have access to additional features that do not depend on reference translations (`string-general` and `linguistic-general`); thus when the metric only has access to the worst MT system as its reference, it can learn to rely on those other corpus-based features. The results of this experiment suggest that learning is important for metrics to exploit pseudo-references properly.

5.5.2 Generalizability of pseudo-reference metrics

The goal of this experiment is to determine whether a metric learned from pseudo-references is effective in practical situations. We consider the case in which the metric is to be used during the internal evaluation of an MT system under development. Using a portion of the parallel text for testing, the developers would have a single reference translation for each sentence; they would also need access to multiple MT systems that are relatively different to create pseudo-references; finally, they need some human-assessed sentences as training examples (such as the NIST MT evaluation datasets). For this experiment, we chose three MT systems to generate the pseudo-references:

Systran,[10] GoogleMT,[11] and a syntax-aware SMT system by Gimpel and Smith (2008). They have been shown to produce good quality translations, and their approaches are relatively different from each other.

As with the earlier generalization experiment, we use the regression-trained metrics to evaluate sentences produced by a diverse set of MT systems. The metric is trained on the NIST 2003 Chinese–English dataset and tested on the Chinese–English datasets from NIST 2002, 2004, 2005, and 2006. We consider two pseudo-reference configurations. In one, no human-produced reference is used (this is labeled as *3 MT Refs*); in the second case, we augment the three pseudo-references with a human reference. The two pseudo-reference configurations are compared against the condition when four human references are available (the same as in Table 4) and when only one human reference is available. Since metrics are more robust when they have access to multiple references, we performed this experiment to see whether pseudo-references may supplement the single human reference.

The resulting correlation coefficients of the metrics using different reference configurations for the four test sets are shown in Table 10. The highest correlation coefficient for each test set is highlighted in bold font, and the coefficients that are not found to be significantly worse according to the statistical test are italicized. For the datasets from 2002 and 2004, using pseudo-references seems to be as informative as using all four human references, and all three metrics have significantly higher correlations than when they use just one human reference. In the later datasets, however, the metrics perform better when they have access to all four human references.

One possible explanation for this discrepancy is that the quality of the MT systems used as pseudo-references is much higher than those of the earlier MT systems being evaluated, while in the later years, they are more similar to the systems they are evaluating. Another possible explanation is that some of the MT systems used as pseudo-references may have been tuned on the earlier NIST data[12] so that their translation outputs would have higher quality than for the later years. Here, we discuss the results for all years because the machine translations used for pseudo-references are still far from perfect and therefore they do not present a conflict with the goal of our experiment, which is to determine whether imperfect references might still be informative for an evaluation metric.

Moreover, because the regression metric is trained on the NIST 2003 dataset, the 2002 and 2004 datasets provide an evaluation in which the testing condition is more similar to the training condition. Tracking the correlation over multiple years shows us how well the metric generalizes to future inputs. Compared to the results of the earlier experiment in Sect. 5.3, when a metric is trained with pseudo-references, it seems to have a shorter shelf-life. As we saw in the previous experiment, the correlation is not as reliable when the pseudo-references are much worse than the population of sentences being evaluated. Thus, a metric that learns to depend on a particular

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

[11] http://www.google.com/language_tools/.

[12] From personal communications, we know that the Gimpel and Smith system used NIST 2003 to select features for their model and a portion of NIST 2004 for parameter tuning for minimum error rate training. We do not know development details of the two off-the-shelf systems.

Table 10 A comparison of metrics using different reference types

Test set	Ref type	Bleu	METEOR	Regression
2002 Chn	1 Human Ref	0.228	0.264	0.288
	3 MT Refs	0.299	0.294	*0.322*
	1 Human Ref + 3 MT Refs	*0.314*	*0.301*	*0.343*
	4 Human Refs	0.283	0.298	**0.337**
2004 Chn	1 Human Ref	0.532	0.519	0.588
	3 MT Refs	0.581	0.564	0.616
	1 Human Ref + 3 MT Refs	0.600	0.577	**0.625**
	4 Human Refs	0.593	0.568	*0.616*
2005 Chn	1 Human Ref	0.460	0.457	0.507
	3 MT Refs	0.382	0.438	0.476
	1 Human Ref + 3 MT Refs	0.431	0.458	0.527
	4 Human Refs	0.489	0.509	**0.565**
2006 Chn	1 Human Ref	0.443	0.459	0.473
	3 MT Refs	0.342	0.412	0.434
	1 Human Ref + 3 MT Refs	0.381	0.438	0.461
	4 Human Refs	0.475	**0.525**	*0.506*

The regression metrics are trained on the NIST 2003 Chinese–English dataset. The NIST Chinese–English datasets from other years are used as test. The highest correlation coefficient for a particular year's dataset is displayed in bold font, and those correlation coefficients that are not significantly different from it (according to the approximate randomized test) are shown in italics

type of MT system to provide the pseudo-references may become less effective as the quality of the reference MT outputs diminishes relative to the newer systems over time. Periodic retraining or adaptive learning methods may help the metric to remain robust.

6 Conclusion

Human judgment of sentence-level translation quality depends on many criteria. Machine learning affords a unified framework to compose these criteria into a single metric. In this paper, we have demonstrated the viability of a regression-based approach to learning the composite metric. Our experimental results suggest that machine learning can successfully combine different individual metrics to create a composite sentence-level evaluation metric that has higher correlations with human judgments than the individual metrics. Moreover, we find that by optimizing against human-assessed training examples, regression methods result in metrics that have better correlations with human judgments even as the distribution of the tested population changes over multiple years. While the training process does introduce some uncertainty because the quality of the resulting metric depends somewhat on the distribution of training examples it saw, we find that it does not have a large impact on the overall effectiveness of the learned metric, especially for the evaluation of higher-quality sentences. Finally, we have presented a method for developing sentence-level MT

evaluation metrics without using human references. We showed that by learning from human-assessed training examples, the regression-trained metric can evaluate an input sentence by comparing it against multiple machine-generated pseudo-references and other target-language resources. We observe that regression metrics that use multiple pseudo-references often have comparable or higher correlation rates with human judgments than standard reference-based metrics. Our study suggests that in conjunction with regression training, multiple imperfect references may be as informative as gold-standard references.

Acknowledgements This work has been supported by NSF Grants IIS-0612791 and IIS-0710695. We would like to thank Kevin Gimpel and Noah Smith for help with their translation system. We would also like to thank NIST for making their assessment data available to us. This paper builds upon earlier work which appeared previously in conferences (Albrecht and Hwa 2007a, b); we thank the reviewers of this paper and the reviewers of the earlier papers for their comments. We are also grateful to Regina Barzilay, Ric Crabbe, Dan Gildea, Alex Kulesza, Alon Lavie, and Matthew Stone for their helpful suggestions on this work.

References

Albrecht JS, Hwa R (2007a) A re-examination of machine learning approaches for sentence-level MT evaluation. In: ACL 2007 Proceedings of the 45th annual meeting of the Association for Computational Linguistics, Prague, Czech Republic, pp 880–887

Albrecht JS, Hwa R (2007b) Regression for sentence-level MT evaluation with pseudo references. In: ACL 2007 Proceedings of the 45th annual meeting of the Association for Computational Linguistics, Prague, Czech Republic, pp 296–303

Al-Onaizan Y, Curin J, Jahr M, Knight K, Lafferty J, Melamed ID, Och F-J, Purdy D, Smith NA, Yarowsky D (1999) Statistical machine translation. Technical report natural language engineering workshop final report. Johns Hopkins University, Baltimore

Banerjee S, Lavie A (2005) METEOR: an automatic metric for MT evaluation with improved correlation with human judgments. In: Proceedings of the workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization, Ann Arbor, Michigan, pp 65–72

Bishop CM (2006) Pattern recognition and machine learning. Springer-Verlag, New York

Blatz J, Fitzgerald E, Foster G, Gandrabur S, Goutte C, Kulesza A, Sanchis A, Ueffing N (2003) Confidence estimation for machine translation. Technical report natural language engineering workshop final report. Johns Hopkins University, Baltimore

Burbank A, Carpuat M, Clark S, Dreyer M, Groves D, Fox P, Hall K, Hearne M, Melamed ID, Shen Y, Way A, Wellington B, Wu D (2005) Final report of the 2005 language engineering workshop on statistical machine translation by parsing. Technical report natural language engineering workshop final report. Johns Hopkins University, Baltimore

Carbonell JG, Cullingford RE, Gershman AV (1981) Steps toward knowledge-based machine translation. IEEE Trans Pattern Anal Mach Intell 3(4):376–392

Corston-Oliver S, Gamon M, Brockett C (2001) A machine learning approach to the automatic evaluation of machine translation. In: Association for Computational Linguistics, 39th annual meeting and 10th conference of the European chapter, proceedings of the conference, Toulouse, France, pp 148–155

Cortes C, Vapnik V (1995) Support-vector networks. Mach Learn 20(3):273–297

Doddington G (2002) Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In: Proceedings of the second conference on human language technology (HLT-2002), San Diego, California, pp 128–132

Dorr BJ, Jordan PW, Benoit JW (1999) A survey of current paradigms in machine translation. Adv Comput 49:2–68

Duh K (2008) Ranking vs. regression in machine translation evaluation. In: Proceedings of the third workshop on statistical machine translation, Columbus, Ohio, pp 191–194

Gamon M, Aue A, Smets M (2005) Sentence-level MT evaluation without reference translations: beyond language modeling. In: 10th EAMT conference, practical applications of machine translation, proceedings, Budapest, Hungary, pp 103–111

Giménez J, Màrquez L (2008) A smorgasbord of features for automatic MT evaluation. In: ACL-08: HLT third workshop on statistical machine translation, Columbus, Ohio, pp 195–198

Gimpel K, Smith NA (2008) Rich source-side context for statistical machine translation. In: ACL-08: HLT third workshop on statistical machine translation, Columbus, Ohio, pp 9–17

Goldberg Y, Elhadad M (2007) SVM model tampering and anchored learning: a case study in Hebrew NP chunking. In: ACL 2007 Proceedings of the 45th annual meeting of the Association for Computational Linguistics, Prague, Czech Republic, pp 224–231

Hastie T, Tibshirani R, Friedman J (2001) The elements of statistical learning. Springer-Verlag, New York

Hovy E, King M, Popescu-Belis A (2002) Principles of context-based machine translation evaluation. Mach Transl 17(1):43–75

Joachims T (1999) Making large-scale SVM learning practical. In: Schöelkopf B, Burges C, Smola A (eds) Advances in kernel methods—support vector learning, MIT Press, Cambridge, pp 169–184

Kauchak D, Barzilay R (2006) Paraphrasing for automatic evaluation. In: HLT-NAACL 2006 Human language technology conference of the North American chapter of the Association for Computational Linguistics, New York, NY, pp 455–462

Koehn P (2004) Statistical significance tests for machine translation evaluation. In: Proceedings of the 2004 conference on empirical methods in natural language processing, Barcelona, Spain, pp 388–395

Kulesza A, Shieber SM (2004) A learning approach to improving sentence-level MT evaluation. In: TMI-2004: Proceedings of the tenth conference on theoretical and methodological issues in machine translation, Baltimore, MD, pp 75–84

Lavie A, Agarwal A (2007) METEOR: an automatic metric for MT evaluation with high levels of correlation with human judgments. In: ACL 2007: Proceedings of the second workshop on statistical machine translation, Prague, Czech Republic, pp 228–231

Leusch G, Ueffing N, Ney H (2006) CDER: efficient MT evaluation using block movements. In: EACL-2006, 11th conference of the European chapter of the Association for Computational Linguistics, proceedings of the conference, Trento, Italy, pp 241–248

Lin C-Y, Och FJ (2004a) Automatic evaluation of machine translation quality using longest common subsequence and Skip-Bigram statistics. In: ACL-04: 42nd annual meeting of the Association for Computational Linguistics, Barcelona, Spain, pp 605–612

Lin C-Y, Och FJ (2004b) ORANGE: a method for evaluating automatic evaluation metrics for machine translation. In: Coling, 20th international conference on computational linguistics, proceedings, Geneva, Switzerland, pp 501–507

Lita LV, Rogati M, Lavie A (2005) Blanc: learning evaluation metrics for MT. In: HLT/EMNLP 2005 Human language technology conference and conference on empirical methods in natural language processing, Vancouver, British Columbia, Canada, pp 740–747

Liu D, Gildea D (2005) Syntactic features for evaluation of machine translation. In: Intrinsic and extrinsic evaluation measures for machine translation and/or summarization, Proceedings of the ACL-05 workshop, Ann Arbor, MI, pp 25–32

Liu D, Gildea D (2006) Stochastic iterative alignment for machine translation evaluation. In: COLING·ACL 2006 21st international conference on computational linguistics and 44th annual meeting of the Association for Computational Linguistics, Proceedings of the main conference poster sessions, Sydney, Australia, pp 539–546

Liu D, Gildea D (2007) Source-language features and maximum correlation training for machine translation evaluation. In: NAACL HLT 2007 Human language technologies 2007: the conference of the North American chapter of the Association for Computational Linguistics, Rochester, NY, pp 41–48

Melamed ID, Green R, Turian J (2003) Precision and recall of machine translation. In: HLT-NAACL 2003: conference combining human language technology conference series and the North American chapter of the Association for Computational Linguistics series, companion volume, Edmonton, Canada, pp 61–63

Owczarzak K, Groves D, Van Genabith J, Way A (2006) Contextual bitext-derived paraphrases in automatic MT evaluation. In: HLT-NAACL 06 Statistical machine translation, Proceedings of the workshop, New York City, pp 86–93

Owczarzak K, Van Genabith J, Way A (2007) Dependency-based automatic evaluation for machine translation. In: Proceedings of SSST, NAACL-HLT 2007/AMTA workshop on syntax and structure in statistical translation, Rochester, NY, pp 80–87

Papineni K, Roukos S, Ward T, Zhu W-J (2002) Bleu: a method for automatic evaluation of machine translation. In: 40th annual meeting of the Association for Computational Linguistics (ACL-2002), Philadelphia, PA, pp 311–318

Quirk C (2004) Training a sentence-level machine translation confidence measure. In: Proceedings of the international conference on language resources and evaluation (LREC-2004), Lisbon, Portugal, pp 825–828

Riezler S, Maxwell JT III (2005) On some pitfalls in automatic evaluation and significance testing for MT. In: Intrinsic and extrinsic evaluation measures for machine translation and/or summarization, Proceedings of the ACL-05 workshop, Ann Arbor, MI, pp 57–64

Snover M, Dorr B, Schwartz R, Micciulla L, Makhoul J (2006) A study of translation edit rate with targeted human annotation. In: AMTA 2006: Proceedings of the 7th conference of the Association for Machine Translation in the Americas, visions of the future of machine translation, Cambridge, MA, pp 223–231

Tillmann C, Vogel S, Ney H, Sawaf H, Zubiaga A (1997) Accelerated DP-based search for statistical translation. In: Proceedings of the 5th European conference on speech communication and technology (EuroSpeech'97), Rhodes, Greece, pp 2667–2670

Uchimoto K, Kotani K, Zhang Y, Isahara H (2007) Automatic evaluation of machine translation based on rate of accomplishment of sub-goals. In: NAACL HLT 2007 Human language technologies 2007: the conference of the North American chapter of the Association for Computational Linguistics, Rochester, New York, pp 33–40

Witten IH, Frank E (2005) Data mining: practical machine learning tools and techniques, 2nd edn. Morgan Kaufmann, San Francisco

Ye Y, Zhou M, Lin C-Y (2007) Sentence level machine translation evaluation as a ranking. In: ACL 2007: Proceedings of the second workshop on statistical machine translation, Prague, Czech Republic, pp 240–247