Part 5: Machine Translation Evaluation

Editor: Bonnie Dorr

Chapter 5.1 Introduction

Authors: Bonnie Dorr, Matt Snover, Nitin Madnani

The evaluation of machine translation (MT) systems is a vital field of research, both for determining the effectiveness of existing MT systems and for optimizing the performance of MT systems. This part describes a range of different evaluation approaches used in the GALE community and introduces evaluation protocols and methodologies used in the program. We discuss the development and use of automatic, human, task-based and semi-automatic (human-in-the-loop) methods of evaluating machine translation, focusing on the use of a human-mediated translation error rate HTER as the evaluation standard used in GALE. We discuss the workflow associated with the use of this measure, including post editing, quality control, and scoring. We document the evaluation tasks, data, protocols, and results of recent GALE MT Evaluations. In addition, we present a range of different approaches for optimizing MT systems on the basis of different measures. We outline the requirements and specific problems when using different optimization approaches and describe how the characteristics of different MT metrics affect the optimization. Finally, we describe novel recent and ongoing work on the development of fully automatic MT evaluation metrics that can have the potential to substantially improve the effectiveness of evaluation and optimization of MT systems.

Progress in the field of machine translation relies on assessing the quality of a new system through systematic evaluation, such that the new system can be shown to perform *better* than pre-existing systems. The difficulty arises in the definition of a *better* system. When assessing the quality of a translation, there is no single correct answer; rather, there may be any number of possible correct translations. In addition, when two translations are only partially correct - but in different ways - it is difficult to distinguish quality. Moreover, quality assessments may be dependent on the intended use for the translation, e.g., the tone of a translation may be crucial in some applications, but irrelevant in other applications.

Traditionally, there are two paradigms of machine translation evaluation: (1) *Glass Box* evaluation, which measures the quality of a system based upon internal system properties, and (2) *Black Box* evaluation, which measures the quality of a system based solely upon its output, without respect to the internal mechanisms of the translation system. Glass Box evaluation focuses upon an examination of the system's linguistic

coverage and the theories used to handle those linguistic phenomena. Individual linguistic components of the system may be examined and be subjected to black box evaluations. This method of evaluation was primarily focused on rule-based expert systems, rather than statistical systems.

Black Box evaluation, on the other hand, is concerned only with the objective behavior of the system upon a predetermined evaluation set. This method of evaluation is only a fair comparison of systems if the systems being tested were both designed to work on data that is of the same character as the evaluation set or, if not, the person testing the systems has the objective of testing robustness across different data types with variations in structure, genre, and style. This method has proved invaluable to the field of machine translation, enabling comparison of systems on the same test sets in order to determine whether a given change to a system is in fact an improvement. The method of actually measuring the performance of a system upon a test set is still a very active research area, and evaluation metrics of this type are the focus of this part.

Within black box approaches both *intrinsic* and *extrinsic* measures are used to assess the accuracy and usefulness of MT output. Intrinsic measures focus on the quality of MT output and often involve quality comparisons between MT output and a set of reference translations that are predetermined to be of high quality. Human intrinsic measures determine quality through human subjective judgments of certain characteristics of the output such as fluency and adequacy. Automatic intrinsic measures use an easily computed sentence similarity measure to produce rankings among MT systems by comparing the corresponding MT output against a fixed set of reference translations. The use of automatic metrics for system optimization of MT systems represents a significant breakthrough in the field of machine translation that has been used heavily in the GALE program. Eight automatic metrics are discussed in this part: BLEU, NIST, METEOR, and WER, PER, GTM, TER, and CDER. These measures have become an essential part of the machine translation research cycle, allowing rapid testing of new features and models, as well as providing a method for the automatic optimization of system parameters.

In contrast to intrinsic measures, extrinsic (task-based) measures are aimed to test the effectiveness of MT output with respect to a specific task. Two examples discussed in this part are extrinsic measures of human performance based on document exploitation task accuracy and measurements of human reading comprehension on machine translated texts. Extrinsic measures are aimed at testing the utility of machine translation. It is this utility testing aspect that served as a crucial foundation for justifying the establishment of the GALE program, wherein measures such as Defense Language Proficiency Test (DLPT) and HTER were first tested on a broad scale.

As HTER has become the primary evaluation criterion for the GALE MT program, a central component of evaluation research is that of relating HTER to automatic measures, with the motivation of establishing an automatic metric that can serve as a surrogate for HTER during system development. In addition, researchers in the GALE program have investigated different optimization techniques for tuning statistical MT systems and system combinations. (For a more elaborate description of system combinations, see the section on System Integration Framework in Part 6 of this book.) Researchers have also focused on the development of improved automatic evaluation metrics for machine

translation, with the goal of improving levels of correlation of metric scores with human judgments of translation quality.

The next three sections provide the definitional foundation for the remaining chapters. Section 5.1.1 focuses on the historical background for the approaches described in this part. Section 5.1.2 discusses evaluation metrics that rely upon human judgments, arguing for the continued use of human judgments in MT evaluations and describing the kinds of judgments that are commonly used. Common practices for generating such judgments are surveyed and efforts to improve the validity and repeatability of human judgments are discussed. Section 5.2.2 examines automatic evaluation metrics that do not require any human interaction (BLEU, NIST, METEOR, WER, PER, GTM, TER, and CDER) and also provides a brief description of the human-mediated standard used in GALE (HTER), relating it to automatic measures.

The following three sections turn to evaluation measures and approaches requiring human intervention. The primary goal of these new metrics is to improve the levels of correlation of metric scores with human judgments of translation quality, especially at the levels of documents and individual segments. Section 5.3 presents two human-in-the-loop extrinsic measures for evaluating the output of machine translation technology: human performance based on document exploitation task accuracy and measurements of human reading comprehension on machine translated texts. Both approaches seek to provide real MT users an alternative, accessible frame of reference for assessment MT engines. Section 5.4 focuses on human post-editing, its challenges and its use in GALE. The entire post-editing process is detailed and the challenges of editor consistency are addressed. Section 5.4.4 provides a detailed overview of the GALE MT evaluations and their results, including a study comparing the different metrics.

The final two sections turn to techniques for using MT evaluation measures in system tuning, approaches to improving on existing measures, and approaches to designing innovative MT measures. Section 5.5.1 describes different optimization techniques (Simplex, Powel's method, etc.) used for tuning statistical MT systems and system combinations. Experiments that highlight different aspects of this optimization and different characteristics of MT metrics are discussed. Finally, Section 5.6 describes significant recent work by members of the GALE research community on developing improved automatic evaluation metrics for machine translation.

5.1.1. Historical Background

Authors: Bonnie Dorr, Mark Przybocki, Matt Snover, Audrey Le, Gregory Sanders, Sebastién Bronsart, Stephanie Strassel, Meghan Glenn

The task of evaluating machine translation quality, like machine translation itself, has a long history (Hutchins 2001). As far back as 1966, in Appendix 10 of ALPAC (1966), experiments were reported with human ratings of intelligibility, as well as the informativeness of a human translation seen after studying the machine translation. One of the legacies of DARPA MT evaluations in the early 1990's has been the use of human subjective judgments to “score” the semantic accuracy and fluency of MT outputs against one or more professionally produced human reference translations. The original hypothesis was that human subjects trained on these scoring tasks could serve as

adequately accurate stand ins for full-fledged professional translators trained to rate the MT outputs according to well-established standards.

Several methods of evaluation using human judgments are frequently employed in the machine translation community. In some cases, the quality of system output is measured directly, such as with human judgments; in other cases, it is measured by performing reading tests or other downstream tasks with the system output, and in still other cases it is measured by calculating the amount of work required to correct the system output.

Two of the most common method human evaluation metrics are fluency and adequacy judgments (White 1994; Callison-Burch 2007). *Fluency* requires a speaker fluent in the target language to judge whether the system output is fluent, regardless of whether content of the output is an accurate translation of the source words.

Adequacy disregards the level of fluency in the system output and, as far as this is possible, measures whether the essential information in the source can be extracted from the system output. The requirements for an annotator of adequacy are stricter than for fluency, as the annotator must be bilingual in both the source and target language in order to judge whether the information is preserved across translation. In practice, an annotator fluent only in the target language could also annotate adequacy using a set of high quality human translations of the source sentence.

Fluency and adequacy are measured separately on each sentence in the system output and are usually judged on a five or seven point scale (Przybocki 2008). They are sometimes averaged to give a single numerical score to a system output. Some studies (Turian 2003; Snover 2006) have shown poor correlation between annotators using this method, bringing into question the reliability of this method. Nevertheless, human evaluation has been used as a baseline by which evaluation metrics are frequently judged. Judgments of semantic adequacy (and related ideas such as understandability or fluency) (Gates 1996; Nubel 1997) have continued to be employed as extremely useful benchmarks for the performance of MT systems and proposed MT metrics, even though the reliability of human judgments remains difficult (Turian 2003).

By the mid 1990's, the results of regular ARPA evaluations of MT led to doubts about the validity and reliability of human ratings of adequacy and fluency of MT output (King 1996) due to compounding factors such as the human evaluators' experience as translators/evaluators and the evaluators' familiarity with the system interfaces. The idea of using multiple *reference translations* for the first *fully automatic* measures was being seriously explored by 1990, when Niessen (2000) investigated a combination of Word Error Rate with sentence-by-sentence selection from multiple human references. Edit-distance metrics have since been explored as measures for MT quality (Frederking 1994; Knight 1994; King 1996).

Post-editing, where the system output is corrected after it is produced, is another common method of measuring translation quality, with more accurate translation requiring less editing and poor translations requiring large amounts of editing. This method suffers as an evaluation metric due to the large amount of work required by human annotators to correct system output, rather than quickly objectively scoring it on an objective scale.

We will see in Section 5.2.3.1 that the notion of post-editing is still relevant in modern measures, e.g., "Human-mediated Translation Error Rate" HTER (Snover 2006), an approach to evaluating machine translation that was developed in and for the GALE program. Although HTER is built on human judgments, its greatest obvious weakness is that it is a purely quantitative metric that weights all errors equally, when in fact some edits, some translation errors, are of trivial importance while others such as some instances of polarity errors (*is* vs. *is not*) can be devastating. On the other hand, human judges are more likely to "forgive trivial" translation errors, or incorporate synonyms in their post edits. This would not be possible in an approach that uses Gold Standard references only.

Task oriented evaluation (White 2000; Jones 2006) is concerned with the effect of machine translation on downstream tasks that use the translation output as input. Recognizing named entities, document categorization, and the effect on reading speed and accuracy are all examples of task oriented evaluation that have been used to measure the quality of machine translation quality. The performance of human evaluators on these tasks is compared across system outputs, as well as to performance when using reference translations. This evaluation method is beneficial in that it measures how useful the system output is for a task rather than how close it is to the reference answers. Translations that are not judged to be perfect according to other measures may still prove to be fully adequate for downstream tasks. Similarly, translations of a certain type may achieve high scores by other measures, but may turn out to be completely inadequate for these tasks, e.g., because of the aforementioned polarity errors. In addition to tasks that require humans to process the system output, one could imagine using the system output in some automatic downstream process, such as information extraction. As an evaluation metric, such automated task evaluations are problematic as they compound errors in the translation process with errors in the downstream automated process, and at best can be used only to measure how useful that translation is for that particular implementation of that downstream automated task.

Automated metrics such as BLEU, NIST, METEOR, WER, PER, GTM, TER, CDER and now HTER (to be presented in Section 5.2) were developed due to the high costs, lack of repeatability, subjectivity, and slowness of evaluating machine translation output using human judgments, and the desire to enable automatic tuning of system parameters (Och 2003). These are based on automated comparisons between MT system outputs and human translations of the same source material, typically looking at *n*-gram matches. Coughlin (2003) examined the nature of the correlation between such *n*-gram based automated metrics and human judgments on a four-point acceptability scale of the same MT outputs, using 124 sentences from a range of European language pairs.

Automated metrics that can somehow give credit for paraphrases and/or synonymy have some intuitive merits. METEOR (Lavie 2004; Banerjee 2005) makes use of stemming and (at least for English) of synonymy apparent in WordNet synsets. TER and HTER allow some rearrangements, scoring a block move as a single edit, which may be particularly useful for target languages that have relatively free word order. Russo-Lassner (2005) explored the possible merits of exploiting paraphrase-like features in MT evaluations. As has been mentioned, not penalizing paraphrases is part of the motivation for using HTER in the GALE program.

Perhaps the closest competing approach to mechanical edit distance measures are those based on concepts of one sort or another. Sanders (2008) presented the evaluations used in the DARPA TRANSTAC program, which are based on purely quantitative scoring of the probability of a successful transfer of open-class content words. Other work based on concepts includes the use of Interlingua Interchange Format (Levin 2000), and predicate argument structures (Belvin 2004).

5.1.2. Human Subjective Judgments

Authors: Gregory Sanders, Mark Przybocki, Nitin Madnani, and Matthew Snover

Two central problems, measuring utility and/or measuring quality, face anyone who sets out to create a new or different evaluation of the outputs of MT systems. Either there is a clear idea of the specific purposes that the translation is to serve and the evaluation focuses on the utility (or acceptability) of the actual MT outputs, and/or there is some idea of what constitutes a better translation with the evaluation focused on the quality of the MT outputs.

The general notion of quality of a translation is, however, somewhat subjective, and the most accepted measure, the Gold Standard, so to speak, is to have truly bilingual human judges compare the source language inputs to the target language outputs and then somehow provide an opinion about the quality of the translation. Such judgments are most often provided on a multi-point scale.

MT systems that take live or recorded spoken inputs must include automatic speech recognition (ASR) abilities, also called speech-to-text (STT). If evaluation of ASR/STT is not desired, textual transcripts may be used as a surrogate input, constituting perfect ASR. Similarly, spoken outputs involve text-to-speech (TTS) abilities, and if evaluation of the TTS is not desired, the textual input to the TTS can be evaluated.

This section of the book focuses on subjective human judgments, whether from bilingual judges who know both the source and the target languages or from monolingual judges who know only the target language. We provide arguments for the continued use of human judgments in MT evaluations, as well as, contrasting approaches using automated metrics, non-subjective counts, or editing-based metrics. We explain what kinds of subjective human judgments are commonly used, survey several ways to generate such judgments, and discuss efforts to improve the validity and repeatability of those judgments.

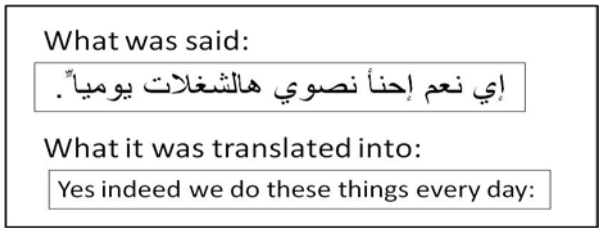


Figure 5.1: A sample source utterance and translation as presented to a bilingual judge.

One type of subjective human judgments involves bilingual human judges reading the source-language input and the corresponding target-language output (see Figure 5.1), then providing a rating on a multi-point scale. As can be seen in Figure 5.1, such ratings require a bilingual judge. The qualities to be rated are most commonly semantic adequacy (is the meaning correct), understandability (is the meaning easy to understand), and fluency (does the MT output have errors that no native speaker would produce). An example of a multi-point scale for semantic adequacy is shown in Figure 5.2. When judging fluency, the source-language inputs are not relevant, so ratings for fluency are usually obtained from monolingual judges.

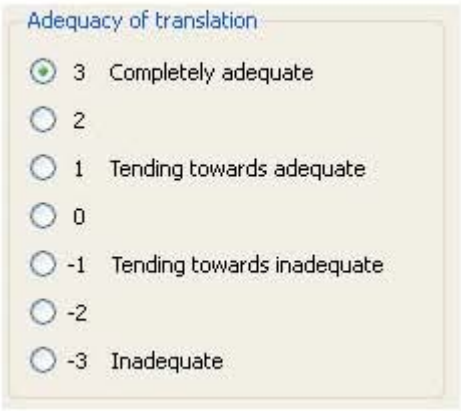


Figure 5.2: Spoken Language Communication and Translation System for Tactical Use

A criticism of using bilingual judges is that their knowledge of the source language is not going to be available to monolingual users, which may hypothetically, lead the bilingual to give a score that is influenced by factors that are of no interest or importance to monolingual users of the translations. We are aware of no real evidence that this is a problem in practice.

We do note that the logistics of using bilingual judges for manual assessment of MT quality are significantly more difficult to arrange as compared to using monolingual judges.

5.1.2.1. Justification for Continued Use of Subjective Human Judgments

There are two strong arguments for using human judgments. The first is that translations are produced for human users, and human judgments are thus the right measure of the qualities of the translation. The second is that human understanding of the real world allows a human judge to assess the practical importance of errors in a translation. For example, if a source language sentence actually means, "There are new

land-mines buried under the road to Baghdad," and the machine translation is, "There are no land-mines buried under the road to Baghdad," then the one small error (of polarity) is terribly important because lives can be at risk. A human judge should recognize and weigh the risk correctly.

Many properties make up the quality of a translation. *Semantic adequacy* (also known as fidelity) is often viewed as the most important, as it answers the question: does the translation have the same *meaning* as the source-language material?

Understandability is also important. A translation may have the completely correct meaning yet be so awkward as to be difficult to understand, noticeably more difficult than the source-language material. Or a source-language idiom may be translated literally and be impenetrable for a target-language hearer/reader. One cannot usefully ask a judge directly whether a translation is understandable: one must find a way to measure or test the judge's understanding.

Finally, fluency is important. The translation may have the correct meaning, be easy to understand, yet could be something that no native speaker of the target language would say or write, for example, “I talks Arabic real OK”, or more severe fluency problems. When judging fluency, the source-language inputs are not relevant, so ratings for fluency are usually obtained from monolingual judges.

Although not discussed here, there are other properties of a translation that one might evaluate via subjective human judgments. For example, the target language output should be appropriate for the intended users. Examples of such problems include ‘Britishisms’ for American users, or vice-versa, female-specific forms when the hearer/reader is male and improper social register (generating language appropriate for speaking to a small child when the person addressed is a distinguished elder).

In the case of human translators using an MT system as an aid, the human translator's proficiency at using the computer interface tools and workstation environment is confounded with measurements of core MT performance (White 1994). This consideration may also be true of human judgments of machine translations, if the judges have difficulty understanding or using whatever system they must use to give their opinions.

Any evaluation of translation involving human judges is inherently also an evaluation of the human judges (King 1996). There are two main concerns with evaluations based on human judgments that must be addressed. The first is that humans will have different opinions and will give different answers for the same assessment. Generally a panel of independent judges is needed to average out those differences. Some of these differing opinions may be attributed to the unavoidable case that judges from different backgrounds tend to weight characteristics of a translation such as syntax “errors” or oddities of style differently (Nubel 1997). Some recent studies (Turian 2003; Snover 2006**)** have shown poor correlation between annotators using subjective judgments, further bringing into question the reliability of these types of human judgments. The second concern is that the evaluation will be affected by whether the judges are familiar with the subject matter and/or sub-language. One can imagine the types of assessment differences that might occur when using judges with no military experience to assess a translation between Marines that reads, “Tell me the identity of the adjacent unit” or using American judges to assess, “Drop that spanner! Pop the bonnet and boot!”

Nevertheless, human evaluation has been used as a baseline by which evaluation metrics are frequently judged. The fact is we have no real substitute for human judgments of translations. Such judgments constitute the reference notion of translation quality.

5.1.2.2. Survey of Common Practices

Several varied kinds of subjective human judgments of machine translation outputs have been tried. In this section, we discuss characteristics that could be used to describe approaches to subjective judgments, survey six approaches, and comment on their apparent problems and strengths. For purposes of giving clear examples of instructions to the judges, this section will assume the target language is English.

All approaches can be seen as a set of choices among a set of alternatives:

(-) Are the judges monolingual or bilingual?
(-) Do the judges have to understand the subject matter?
(-) Is there a reference for the judges? A reference could be either the source-language material or else one-or-more target language human translations.
(-) What characteristic(s) are to be measured: fluency, semantic adequacy, relative quality, acceptability?
(-) What type of judgment is to be made: a yes/no judgment (as of acceptability), a judgment on some sort of numeric scale (allowing correlation/regression analyses), a categorical judgment, or a preference judgment (possibly even a rank-ordering).
(-) How to analyze the judgments statistically or use them for formative or summative evaluation purposes.
(-) Is the goal to evaluate the translation of some particular text/audio item, or is the goal to evaluate the overall performance of the machine translation system? In practice, this particular question is about the use to be made of the evaluation results rather than the evaluation approach to be used.

	How do you judge the fluency of this translation? It is:
5	Flawless English
4	Good English
3	Non-native English
2	Disfluent English
1	Incomprehensible

Figure 5.3: One example of a multi-point fluency scale, with anchor text for each value

5.1.2.2.1. Fluency ratings from literate monolingual speakers of the target language.

For proper fluency assessments the judge should have access to only the translation being assessed, and not the original source data or a reference translation. The question put to the judges is typically along the lines of, “Is this good English?” It is common to

ask the judge to choose a rating from a multi-point scale. Typically, the choices have some anchor value or text, as in Figure 5.3, which displays the scale (Linguistic Data Consortium 2005) that was used in annual NIST OpenMT evaluations and in a meta-evaluation (Callison-Burch, *et al*. 2007) that was done for the ACL-2007 Workshop on Statistical Machine Translation.

The most negative anchor text in the example above may be construed as conflating the notion of comprehensibility with that of fluency, an effect that could be lessened if the anchor text were instead *"Word salad / This is not English."*

The main problem encountered with subjective human judgments of fluency is that understandability or comprehensibility of the translation enters into the picture. This is a problem if the judges are not familiar with the subject matter, and is a particularly acute problem if the source-language material is disfluent, as is often the case with spoken language, or unstructured texts such as web data.

This approach can be characterized with respect to our bullet list of choices. Fluency judgments can be given by monolingual judges and it is helpful if the judges understand the subject matter. Fluency judgments do not require a reference translation (or the source-language material). The quality to be judged is fluency. (Is this good English?) The judgments are usually given as categorical rankings, but the judgments are often analyzed as numeric - as seen in the numbers that begin the anchor values in our example.

Analyzing these judgments as categorical data is favored unless the judges are explicitly presented with the numeric interpretation of their choices and informed that those numbers are the data that will be analyzed and that the numeric values are presumed to be equally spaced.

5.1.2.2.2. Using fluency ratings from literate monolingual speakers of the target language as a proxy for accuracy ratings.

Wilks (2008) demonstrated that reasonably fluent MT output is also usually reasonably correct in meaning. For that reason, ratings of fluency by a monolingual judge often correlate with judgments of semantic adequacy sufficiently that (strange as the methodology might be) fluency judgments have some power to predict ratings of semantic adequacy.

In contrast, human language learners typically produce translations that are fluent even when the meaning is wrong, perhaps because a human translator (in contrast to a machine translation system) may construct a plausible meaning. A truly bilingual judge can detect such errors.

It is not known whether the correlation (between fluency and adequacy judgments of machine translation outputs) is a result of fluency judges assigning lower fluency scores when the MT output contradicts human knowledge of the real world.

With respect to the bullet list of choices, this approach is like the previous approach, but the caveats mentioned for this approach turn on the use of monolingual judges and the effects of the judges' understanding of the subject matter. Note that although the judgments are of fluency, the quality to be measured, indirectly, is semantic adequacy. The judgments are usually given as categorical choices, but the judgments are often

analyzed as numeric. It would probably be most appropriate to analyze these data as categorical and to attempt to validate this approach by showing some systematic relationship between the fluency judgments and some independent measure of semantic adequacy. The authors are not persuaded that this validity will always exist.

5.1.2.2.3. Semantic adequacy from truly bilingual judges.

This is widely regarded as the Gold Standard for assessment of MT quality. In addition to the translation being assessed, a bilingual judge will have access to the original source text and optionally a reference translation. The question put to the judge usually asks for a categorical rating based on the meaning contained in the translation as compared with the source text. It is necessary to use judges who understand the subject matter sufficiently for such judgments to be valid. In addition, for bilingual judges, their proficiency and native language is also important.

Typically, the human judgments are given on a multi-point scale, such as that in Figure 5.3.

A problem with using such a scale for any type of subjective human assessment is that the level of agreement between judges is typically not high, unless one counts the agreements as including the assessments that disagree by one category on a five-point scale, or by up to two categories when using a seven-point scale. This problem of inter-judge agreement can be dealt with by averaging over a panel of several judges.

One can treat the data as categorical and present the results for each judge separately, looking to see whether the judges agree about which items are better or worse. Or, if the scores have a valid numeric interpretation, then for comparisons where each judge gives scores on a set of many items, and all judges score the same set of items, one can normalize the data from each judge separately to a z-statistic (mean -0.0 and standard deviation = 1.0).

However, if the goal is an absolute measurement of quality, or to compare performance between a current evaluation and a previous evaluation, normalizing across judges is of course not relevant. For comparisons to measure progress between performance on a current evaluation and performance on a previous evaluation, especially if the panel of judges changes from one evaluation to the next, one falls back on trying to hold the evaluation procedures as constant as possible - there is no statistical magic bullet in this situation. With a panel of several judges, more robust statistics such as a trimmed mean (say, counting only the middle five judges from a panel of nine), the median, or the inter-quartile range may be used.

If the goal is to assess fitness for some purpose, it may be possible to find a judge, or judges, who represent the intended users and whose assessments are known a-priori to accomplish that goal.

If a machine translation system or the translation itself is to be assessed, these judgments will be used directly. But, if an automated metric for machine translation quality must be assessed, that assessment is usually based on the degree of correlation between the scores from the automated metric and these ratings of semantic adequacy from bilingual judges.

Returning to our bullet list of characteristics of assessment techniques, these judgments are from truly bilingual judges and it is important that they understand the

subject matter in order for their judgments to reflect the quality of the translation. Judgments of semantic adequacy require a reference translation (or the source-language material). The quality to be judged is semantic adequacy (i.e., whether the translation conveys the meaning). The judgments are usually given as categorical rankings, but may be given as numeric, even as an effectively real-number value such as percentage. If the judgments are given on a categorical scale, the data should be analyzed as categorical unless the judges are presented with the numeric interpretation of their choices and informed that the judgments will be analyzed as those numbers.

5.1.2.2.4. Accuracy ratings from literate monolingual speakers of the target language who consult one or more careful human translations.

This is a widely used method for assessing the semantic adequacy of translations, due in part to the ease of obtaining monolingual judges. In addition to the translation being assessed, a monolingual judge will have access to one or more reference translations. The question put to the judge usually asks for a categorical rating based on the meaning contained in the translation as compared with the reference translation. An example is the following scale, Figure 5.4, in which "gold-standard" means a careful human reference translation.

	How much of the meaning expressed in the gold-standard translation is also expressed in the target translation?
5	All
4	Most
3	Much
2	Little
1	None

Figure 5.4: Another scale for semantic adequacy

In practice, this is popular because it tends to be more convenient and less expensive than the approach just discussed using bilingual judges. However, this approach has its limitations. When comparing a translation against a single human reference translation it relies on the reference translation as being perfect, a problem that is worse if the reference translation gives no alternatives for ambiguous passages. Everything mentioned elsewhere in this section about the need for the judges to understand the subject matter and sub-language applies equally to human translators. So, errors introduced by a single human translator will cascade through these subjective assessments.

Given somewhat difficult source-language material, and multiple human reference translations, one commonly finds that the human translators have disagreed about the meaning of the source material. Many such disagreements can only be understood by someone who understands the source language.

The balance switches, and criticisms evaporate, when one has several *completely independen*t careful human translations, because they may bring out ambiguities that

would otherwise have escaped the notice of a judge, or may fortuitously involve at least one translator who is sufficiently familiar with the subject matter as to understand the source material correctly. These advantages of multiple source language experts (the translators) independently giving careful consideration to the meaning of the source material can overcome all the edge of having bilingual judges who will work independently (and thus not take advantage of insights to which only one judge arrives) and who may ponder the source material less thoroughly than careful translators due to focusing on only the errors of the machine translation. Of course, introducing several reference translations into the assessment process may increase the cognitive burden placed on the human judge.

With respect to the bullet list of characteristics of assessment techniques, these judgments are from monolingual judges, but in all other respects have the same characteristics as the version using bilingual judges.

5.1.2.2.5. Informativeness ratings by literate monolingual speakers of the target language.

The essence of this approach is to have the judge make a thorough attempt to understand the translation (perhaps by asking the judge to rewrite it so that it is completely clear), and then *afterwards* show the judge a careful human translation and have the judge rate how informative the reference human translation turned out to be. For example, "Did the reference translation change your understanding of what the passage meant?" This can be rated on a multi-point scale. This approach has been around since the earliest days of machine translation research: it was first described in the ALPAC report (ALPAC 1966). The crucial point of this approach is that the judge must thoroughly consider the translation and arrive at a frozen interpretation thereof before being allowed to see the reference translation; one wants to guard against the judge who realizes only in retrospect (after having seen the reference) that the meaning was obvious and then says the reference was not informative. Although one could simply ask the judge to arrive at a frozen understanding before viewing the reference, it is probably useful to force this issue. For example, one might ask the judge, before seeing the reference, to edit the translation to make it clear and fluent, an editing approach used by Callison-Burch *et al*. (2007).

The key advantage of obtaining ratings of the informativeness of a human translation is that it is a measure not just of semantic adequacy but also of how well a monolingual speaker of the target language understood the machine translation output before seeing a reference translation.

If judges are truly bilingual, a slightly different version of this evaluation can be performed, with the informativeness step showing the judge the source material (instead of a reference translation). However, this version gives up the advantage of having judges who are naive about the characteristics of the source language.

Returning to our bullet list of characteristics of assessment techniques is particularly interesting here because this approach is so different from those described so far. These judgments can be from either monolingual or bilingual judges. It is important for the judges to understand the subject matter in order for their judgments to reflect the quality

of the translation. Judgments of informativeness require a reference human translation or translations. The quality to be judged is semantic adequacy (i.e., whether the translation conveys the meaning), but this quality is measured in an interestingly indirect way that encompasses the understandability of the translation. Although the judgments could be given as a yes/no response, it is far more useful to obtain responses where the categories get at the degree of change in the judges understanding and/or ask the judge whether the problem was with the meaning of the MT output or it's comprehensibility.

5.1.2.2.6. Preference ratings by a literate monolingual speaker of the target language or from bilingual judges.

Often used as a quick approximation for improvement in the development work of system building, this approach presents a monolingual judge with a pair of translations and the corresponding reference translation or presents a bilingual judge with the pair of translations plus the source material. The question put to the judge is along the lines "Which translation do you prefer?"

If one wishes to rank-order multiple translations of the same material, one may use pair-wise preference judgments to build a rank-ordering across several systems. Alternatively, one can present all the versions together and ask the judge to rank-order them.

For a formative evaluation, one might wish to ask the judges to indicate why they prefer one translation over the other, although the judge may not be able to provide an accurate answer.

Our bullet list of characteristics of assessment techniques is also interesting here. These judgments can be from monolingual or bilingual judges. It is important for the judges to understand the subject matter. Monolingual judges require one or more high quality reference human translations and bilingual judges should be provided with the source-language material (either instead of, or in addition to, the human reference translations). The quality to be judged should be overall quality - it is unreasonable to believe judges could isolate quality such as semantic adequacy when making preference judgments. The judgments are preferences which could be implemented in a full pair-wise fashion, comparing two system translations at a time, or in theory, judges could be presented with several translations and asked to rank order their preferences. Rank-ordering may be more burdensome and time-consuming but would avoid the A>B, B>C , and C>A scenario that one might find with pair-wise judgments. There is reason to believe pari-wise rankings will be performed more reliability than directly rank ordering several translations. The data should be analyzed as ranks.

Chapter 5.2 Automatic and Semi-Automatic Measures

Authors: Matthew Snover, Nitin Madnani, Bonnie Dorr, and Richard Schwartz

5.2.1. Introduction

Automatic machine translation evaluation metrics were developed due to the high costs, lack of repeatability, subjectivity, and slowness of evaluating machine translation output using human judgments, and the desire to enable automatic tuning of system parameters (Och 2003). These automatic measures generally judge the quality of machine translation output by comparing the system output, also referred to as candidates or hypotheses, against a set of reference translations of the source data.

The quality of these evaluation metrics is usually measured by determining the correlation of the scores assigned by the evaluation metrics to scores assigned by a human evaluation metric, most commonly fluency and adequacy. In some cases, single scores for systems averaged over entire test sets are compared, and in others document or even sentence level scores are compared. The use of Spearman versus Pearson correlation coefficients also varies within the field. An alternate, although rarely examined, method of comparing automatic evaluation metrics is to examine their effect upon parameter tuning of the MT system. It has not been established that an MT system, which has its parameters tuned to maximize the performance on a metric that correlates better with human judgments, will actually produce better MT output. Such a study is hampered by both the varying parameters of MT systems which allow different aspects of different MT systems to be controlled in different ways as well as variance in parameter optimization methods. Such a study is necessary though for a full understanding of the automatic evaluation of machine translation.

This section details some of the more common automatic evaluation metrics that are used for MT evaluation and parameter optimization, as well as one semiautomatic variant. Word Error Rate (WER) (Section 5.2.2.1) the standard metric of Automatic Speech Recognition performance and one of the first automatic metrics applied to machine translation. BLEU (Section 5.2.2.2), a widely used precision oriented measure that counts the *n*-gram matches between the MT output and the Gold Standard references. (NIST is a variant of BLEU that is weighted according to *n*-gram informativeness.) METEOR (Section 5.2.2.3), a recall oriented measure that utilizes stemming and synonymy to better align the MT output the Gold Standard references. Translation Edit Rate (TER) (Section 5.2.2.4), an extension of the Word Error Rate measure WER (and its variants, MWR, PER and GTM) that allows the movement of word sequences within the MT output. All of these automatic measures rely upon a comparison of the MT output to a set of Gold Standard references, which represent only a small sample of the set of possible correct translations. Human-mediated TER (HTER) (Section 5.2.3.1), which is used as the primary criteria for evaluating the MT systems in GALE, addresses this lack of Gold-Standard references by having human annotators create new references that are as similar as possible to the MT system output, and evaluating the system output with TER using these targeted references.

These automatic metrics represent just a small sampling of the automatic evaluation metrics that been proposed in the machine translation research community and were chosen due to their prominence and use in the GALE research community.

5.2.2. Automatic Measures

5.2.2.1. Word Error Rate (WER MWER) and Position-Independent Error Rate.

One of the first automatic metrics used to evaluate MT systems was Word Error Rate (WER), which is the standard evaluation metric for Automatic Speech Recognition. WER is computed as the Levenshtein distance (Levenshtein 1966) between the words of the system output and the words of the reference translation divided by the length of the reference translation. The Levenshtein distance is computed using dynamic programming to find the optimal alignment between the MT output and the reference translation, with each word in the MT output aligning to either 1 or 0 words in the reference translation, and vice versa. Those cases where a reference word is aligned to nothing are labeled as *deletions*, whereas the alignment of a word from the MT output to nothing is an *insertion*. If a reference word matches the MT output word it is aligned to, this is marked as a *match*, and otherwise is a *substitution*. The WER is then the sums of the number of substitutions (S), insertions (I), and deletions (D) divided by the number of words in the reference translation (N) as shown in Equation (5.1).

$$WER = \frac{S+I+D}{N} \tag{5.1}$$

MWER (Multi-Reference WER) (Nießen 2000) – the application of WER to more than one reference translation – refers to the minimum of the WER scores between the MT output and each reference. In essence, MWER is the WER between the MT output and the closest reference translation. While this allows WER to be used with multiple references, the references are not combined in any fashion and are not truly exploited by the metric.

Unlike speech recognition, there are many correct translations for any given foreign sentence. These correct translations differ not only in their word choice but also in the order in which the words occur. WER is generally seen as inadequate for evaluation for machine translation as it fails to adequately combine knowledge from multiple reference translations and also fails to model the reordering of words and phrases in translation.

Position-independent Error Rate or (PER) (Tillmann 1997) is an attempt to address the word-ordering limitation of WER by treating the reference and hypothesis as bags of words, so that words from the hypothesis can be aligned to words in the reference regardless of position. Because of this the PER OF an MT output is guaranteed to be lower than or equal to the WER of the MT output. This variant has the disadvantage of being unable to distinguish a correct translation from one where the words have been scrambled.

5.2.2.2. BLEU

BLEU (Bilingual Evaluation Understudy) (Papineni 2002) is the current standard for automatic machine translation evaluation. Like MWER, a key characteristics of BLEU is its direct exploitation of multiple references. The BLEU score of a system output is

calculated by counting the number of *n*-grams, or word sequences[1], in the system output that occur in the set of reference translations. BLEU is a precision-oriented metric in that it measures how much of the system output is correct, rather than measuring whether the references are fully reproduced in the system output. BLEU could be gamed by producing very short system outputs consisting only of highly confident *n*-grams, if it were not for the use of a brevity penalty which penalizes the BLEU score if the system output is shorter than the references.

$$p_n = \frac{\sum_{c \in \{can\}} \sum_{n-gram \in c} Cnt_{clip}(n-gram)}{\sum_{c' \in \{can\}} \sum_{n-gram' \in c'} Cnt(n-gram')} \tag{5.2}$$

$$BP = \begin{cases} 1, & if\ c > r; \\ e^{\left(1-\frac{r}{c}\right)}, & if c < r. \end{cases} \tag{5.3}$$

$$BLEU = BP \cdot \exp\left(\sum_{n=1}^{N} w_n log p_n\right) \tag{5.4}$$

Equation (5.2) shows the computation of the BLEU precision scores for *n*-grams of length n, where *Can* are the sentences in the test-corpus, $Cnt(n-gram)$is the number of times an *n*-gram occurs in a candidate, and $Cnt_{clip}(n-gram)$is the minimum of the unclipped count and the maximum number of times it occurs in a reference translation. Equation (5.3) shows the calculation of the BLEU brevity penalty, where c is the length of the candidate translation, and r is the length of the reference translation. These terms are combined, as shown in Equation (5.4), to calculate the total BLEU score, where N is typically 4, and w_n is usually set to $1/N$.

Several variants of the BLEU measure are commonly used that primarily differ in the calculation of the number of reference words used to calculate the brevity penalty. The IBM version of BLEU uses the average lengths of the references, while the NIST version of BLEU uses the shortest reference to calculate brevity penalty. The NIST automatic evaluation metric[2], not to be confused with the NIST version of BLEU, is a variant of the BLEU metric that differs in that it uses the arithmetic mean of the *n*-gram counts, rather than the geometric mean used in BLEU. More importantly, the NIST metric does not treat all *n*-grams equally, but weights them according to their informativeness. The frequency of an *n*-gram sequence is calculated from the set of reference translations, rather than from an external corpus.

Since its introduction, BLEU has become widespread in the machine translation community and is the most commonly reported evaluation metric. Several shortcomings of the BLEU evaluation metric have been brought forth by the measure's critics (Turian 2003; Lavie 2004; Callison-Burch 2006). One of the primary critiques of BLEU is absence of recall in its formulations. In addition, BLEU was designed for, and has been shown to work best when used on, large test corpora, such that the scores are averaged over many sentences. BLEU scores of individual sentences are not considered reliable. Other shortcomings of BLEU are the lack of synonym matching and the inability to detect

[1] A maximum length of four words is common.

[2] http://www.itl.nist.gov/iad/mig/tests/mt/doc/ngram-study.pdf

multiple proper word orders. In short, translation quality - especially semantic quality - -- is not detected by the BLEU measure.

A number of new automatic evaluation measures for machine translation have been proposed in recent years to compensate for the perceived failings of the BLEU scoring measure. These measures all fundamentally deal with the notion of string matching between reference translations and hypothesized translations. The following sections describe several of these more recent measures in detail.

Despite these criticisms, BLEU remains the most commonly used automatic metric both for the optimization of system parameters and for final evaluation of the quality of an MT system. The use of the BLEU metric has driven development in the MT research community, and it is now the automatic evaluation metric against which all new metrics are compared.

5.2.2.3. METEOR

METEOR (Metric for Evaluation of Translation with Explicit ORdering) (Banerjee 2005) is an evaluation specifically designed to address several observed weaknesses in BLEU. METEOR is a recall-oriented metric, whereas BLEU is generally precision-oriented metric[3]. Unlike BLEU which only calculates precision, METEOR calculates both precision and recall, and combines the two, as shown in Equation (5.5), with a large bias towards recall, to calculate the harmonic mean[4]. In more recent work (Lavie 2007), higher correlations with human judgments were obtained by optimizing the parameters of the harmonic mean for specific target languages.

$$F_{mean} = \frac{P \cdot R}{\alpha P + (1-\alpha) R} \tag{5.5}$$

METEOR uses several stages of word matching between the system output and the reference translations in order to align the two strings. The matching stages are as follows:

1. **Exact matching**. Strings which are identical in the reference and the hypothesis are aligned.
2. **Stem matching.** Stemming is performed, so that words with the same morphological root are aligned.
3. **Synonymy matching**. Words which are synonyms according to WordNet (Fellbaum 1998) are aligned.

In each of these stages, only words that were not matched in previous stages are allowed to be matched. Only unigrams, single words, are compared for matches.

[3] The brevity penalty in BLEU addresses this issue by penalizing short translation which BLEU would otherwise be unfairly biased towards. Without the brevity penalty, BLEU would be purely a precision-oriented metric.

[4] The default parameters for the harmonic mean set $\alpha = .9$. It is because of these parameters that METEOR is a recall oriented metric.

Precision in METEOR is defined as number of matches divided by the number of words in the system output, and recall is defined as the number of matches divided by the number of words in the reference.

In addition to the F_{mean}, METEOR also uses a fragmentation penalty to bias the score against system outputs that have many short sequences of consecutive matches, called chunks. Fragmentation is calculated as the number of chunks divided by the number of unigram matches. The fragmentation is calculated as shown in Equation (5.6), with default parameters of $\beta = 3.0$ and $\gamma = .5$.

$$Pen = \gamma \cdot frag^{\beta} \tag{5.6}$$

This fragmentation penalty causes METEOR to correctly penalize "word salad" MT output that would be allowed under the PER metric, and is an essential portion of the METEOR scoring metric.

The final METEOR score is calculated as: $score = (1 - Pen) \cdot F_{mean}$.

Unlike BLEU, METEOR does not penalize longer answers and incorporates a level of linguistic knowledge in the form of its stem and synonym matching allowing it to identify equivalences between the MT output and the reference translation that would ignored by these earlier measures. METEOR lacks one of BLEU's key features however: the direct exploitation of multiple references, as METEOR cannot combine knowledge from multiple references into its score, but rather, when multiple references are available, METEOR selects the reference translation for each segment that gives the best METEOR score. Furthermore, METEOR'S ability to handle variability via stemming and synonyms already reduces the expected gain from comparing against multiple references simultaneously, mitigating the effect of this limitation.

The highly recall based measure though can be exploited by the inclusion of additional highly likely words (such as "the" in English) in the MT output, giving higher scores to outputs with these additional padded words - although such behavior is not typically exhibited by modern machine translation systems. The benefits of a precision oriented metric such as BLEU versus a recall oriented metric such as METEOR must be tested experimentally in terms of correlation with human judgment.

5.2.2.4. General Text Matcher (GTM), Translation Edit Rate (TER) and CDER

As discussed in Section 5.2.2.1, WER (in its standard form) is viewed as inadequate for machine translation evaluation due to its failure to utilize multiple reference translations and its failure to model the reordering of words and phrases in a translation. This section discusses several recent evaluation metrics that attempt to address this latter failing.

The General Text Matcher (GTM) (Turian 2003) evaluation metric attempts to model the movement of phrases during translation by using the maximum matching size to compute the quality of a translation. It finds the longest sequences of words that match

between the hypothesis and the reference[5]. The size of the matches is defined in Equation (5.7), where *M* is the set of matches found. This formulation can be generalized to other exponents, and in experimental results (Turian 2003) the metric appears to work best when $size(M) = \sum_{r \in M} length(r)$. The precision and recall are computed as the size of the matches divided by the length of the system output or the reference, respectively. The score of the GTM is the harmonic mean, F-score, of precision and recall.

$$size(M) = \sqrt{\sum_{r \in M} length(r)^2} \tag{5.7}$$

GTM can also be used with multiple references by concatenating the references together, and not allowing a match to cross the boundary between references.

Like BLEU (and also TER, discussed below), GTM does not incorporate any linguistic knowledge, and only considers words in the MT output and the references as matching if they are identical. The typical definition $size(M)$also generates several interesting properties that are not present in TER however. If a single word in the translation is incorrect, the affect on TER is the addition of a single edit, while in GTM the effect depends upon the location of the error in the sentence. An error in the middle of the sentence has the largest effect, resulting in two matches of equal size, while an error closer to either end of the sentence has a minimal effect, only reducing the length of the match by 1 if the error is the first or last word. Whether errors in the middle of the sentence are of much greater importance that errors at the beginning or end of the sentence, rather than such errors being of equal importance, as in TER, has not been adequately studied.

Translation Error Rate (TER) (Snover 2006) addresses the phrase reordering failing of WER by allowing block movement of words, also called *shifts*, within the hypothesis as a low cost edit, a cost of 1, the same as the cost for inserting, deleting or substituting a word. While a general solution to WER with block movement is NP-Complete (Lopresti,1997), TER addresses this by using a greedy search to select the words to be shifted, and as well as further constraints on the words to be shifted. These constraints are intended to simulate the way in which a human editor might choose the words to shift.

The shifting constraints used by TER serve to both reduce the computational complexity of the model and better model the quality of translation. Examining a larger set of shifts, or choosing them in a more optimal fashion might result in a lower TER score, but it would not necessarily improve the ability of the measure to determine the quality of a translation. The constraints used by TER are as follows:

1. Shifts are selected by a greedy algorithm that selects the shift that most reduces the WER between the reference and the system output.
2. The sequence of words shifted in the system output must exactly match the sequence of words in the reference that it is being shifted to.

[5] Because the solution to this is conjectured to be NP-hard, a greedy search is used to iteratively select the longest sequences. The authors claim that this obtains the optimal solution in 99% of cases.

3. The words to be shifted must contain at least one error, according to the WER, before being shifted. This prevents the shifting of words that currently correctly matched.
4. The matching words in reference that are being shifted to must also contain at least one error. This prevents shifting to align to words that already correctly aligned.

When TER is used in the case of multiple references, it does not combine the references, but scores the hypothesis against each reference individually. The reference with which the hypothesis has the fewest number of edits is deemed the closest reference, and that number of edits is used as the numerator for calculating the TER score, as is done in MWER. Rather than use the number of the words in the closest reference as the denominator, TER uses the average number of words across all of the references. Thus, the equation for the TER score, where SUB, INS, DEL and $SHIFT$ are the number of substitutions, insertions, deletions and shifts, respectively, and $\bar{N}$ is the average number of reference words, is shown in the following equation.

$$TER = \frac{SUB+INS+DEL+SHIFT}{\bar{N}} \tag{5.8}$$

TER cannot exploit multiple references as is done in BLEU nor does it incorporate external linguistic knowledge, e.g., synonyms, as is done in METEOR. These failings can be addressed through the use of targeted references - references created by humans specifically for a particular machine translation output - changing it from an automatic metric (TER) to a semi-automatic metric (HTER) as is discussed in Section 5.2.3.

Other automatic metrics for MT evaluation exist that follow the general formulation as TER, but address the complexity of shifting in different ways, such as CDER (Leusch 2006) (Cover Disjoint Error Rate). CDER exploits the fact that the number of blocks in a sentence is equal to the number of gaps among the blocks plus one. Thus, the block movements can equivalently be expressed as *long jump* operations that jump over the gaps between two blocks. The costs of a long jump are considered constant. The blocks are read in the order of one of the sentences. These long jumps are combined with the "classical" Levenshtein edit operations, namely *insertion, deletion, substitution*, and the zero-cost operation *identity*. The resulting long *jump distance* d_{LJ} gives the minimum number of operations which are necessary to transform the candidate sentence into the reference sentence. Like the Levenshtein distance, the long jump distance can be depicted using an alignment grid as shown in Figure 5.5. Here, each grid point corresponds to a pair of inter-word positions in candidate and reference sentence, respectively. d_{LJ} is the minimum cost of a path between the lower left (first) and the upper right (last) alignment grid point which covers all reference and candidate words. Deletions and insertions correspond to horizontal and vertical edges, respectively. Substitutions and identity operations correspond to diagonal edges. Edges between arbitrary grid points from the same row correspond to long jump operations. It is easy to see that d_{LJ} is symmetrical.

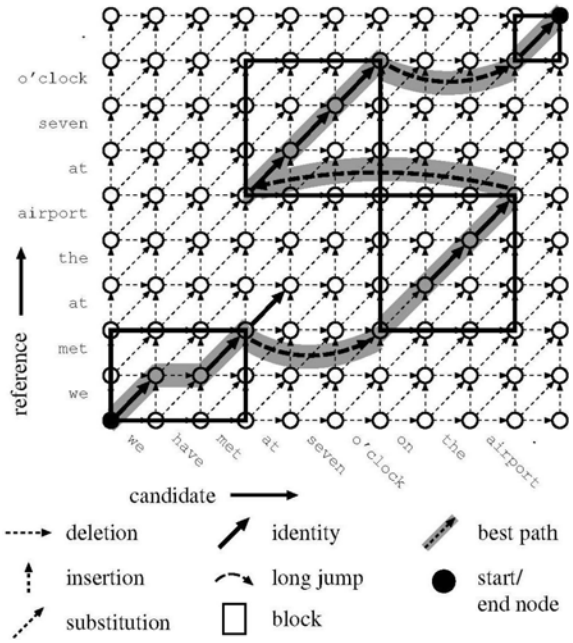


Figure 5.5: Example of a long jump alignment grid All possible deletion, insertion, identity and substitution operations are depicted. Only long jump edges from the best path are drawn.

While finding an optimal path in a long jump alignment grid is an NP-hard problem, CDER uses an approach which has a suitable run-time, while still maintaining completeness of the calculated measure. The idea of the proposed method is to drop some restrictions on the alignment path.

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

CDER can thus be seen as a measure oriented towards *recall*, while measures like BLEU are guided by *precision*. The CDER is based on the $\overline{CD}CD$ *distance*[6] introduced by Lopresti and Tomkins (Lopresti 1997). It can be shown that the problem of finding the optimal solution can be solved in $O(I^2 \cdot J)$ time, where I is the length of the candidate sentence and J the length of the reference sentence. Using a modification of the Levenshtein algorithm, the time complexity can be further reduces to $O(I \cdot J)$ (Leusch 2006).

[6] *C* stands for *cover* and *D* for disjoint.

5.2.3. Semi-automatic Measures

5.2.3.1. HTER

HTER Human-mediated Translation Error Rate) (Snover 2006) is a human-in-the-loop variant of TER that has also been used to evaluate machine translation systems. HTER requires the use of mono-lingual human annotators who create references that are targeted to a particular system output. Targeted references are crafted by changing the system output with a minimal number of edits so that is both fluent and preserves the meaning of the other reference translations. Because a minimal number of edits are used to correct the system output, creating a targeted reference can be thought of as selecting from the set of all possible references the one which is closest, as measured by TER, to the system output. Targeted references could be used with any other automatic evaluation metric that uses reference translations. The use of targeted references has been shown to increase the correlation of automatic metrics with subjective human judgments, and can be seen as a method of addressing the sparsity of reference translations.

Because targeted references are tailored for each particular system output, they cannot be reused for system output from different systems or even output from a different version of the original machine translation system. Because targeted references cannot be reused and the need for human annotators to create targeted references, HTER is completely unsuited as a purely automatic machine translation evaluation metric. Viewed as a type of human judgment though, HTER is an expensive, but fine grained measure of translation quality, and has been used as the primary metric of MT performance in the GALE program. The full details of HTER as implemented in the GALE program are discussed in Section 5.4.

Although HTER is built on human judgments, its most obvious weakness is that it is a purely quantitative metric that weights all errors equally, when in fact some edits, some translation errors, are of trivial importance while others such as some instances of polarity errors (is vs. is not) can be devastating. Such a problem could possibly be addressed by using an automatic measure other than TER to compute the resulting error between the targeted reference and the hypothesis, although this leaves the question of which errors to more heavily. The answer to this is likely to depend heavily on the intended use of the translation[7].

Chater 5.3 Tasks and Human-in-the-Loop Measures Document-Exploitation Task Accuracy

5.3.1. Introduction

Authors: Clare Voss and Doug Jones

[7] We note that, in recent variations of HTER, the evaluator is able to specify the ``weight'' of different types of edits, thus enabling a task-oriented evaluation approach, where certain errors may be penalized more heavily than others.

In addition to the use of human judgments for comparison-based approaches to MT evaluation, humans are also central to extrinsic measures that have been adopted for the purpose of evaluating the output of machine translation. Two examples are measurements of human performance based on document exploitation task accuracy and measurements of human reading comprehension on MT texts. Both approaches seek to provide real MT users with an alternative, accessible frame of reference for assessment of MT engines. Since machine translation may distort or delete portions of the original content, task developers in both approaches must distinguish assessments of:

- Mistakes in the MT text proper, that can be measured indirectly by human subjective judgments, as described in Section 5.1.2 or directly by intrinsic metrics with automated comparisons to human reference translations, as described in Section 5.2.
- Human response errors - whether caused by inherent difficulty of the text-handling task, human performance factors (such as fatigue, inadequate task training, or other individual differences), or mistakes in the MT text.

Developers of task metrics must: *first* establish the ground truth or answer set for the task and document collection, which entails understanding its foreign language content and typically building reference translations of the collection, *second*, after developing the task protocol and establishing the experimental design and statistics, train the subjects and conduct the task under experimental conditions with MT documents, and *third*, assess the accuracy of task responses in relation to information conveyed in the test documents.

5.3.2. Document Exploitation

Author: Clare Voss

5.3.2.1. Introduction

Document collectors in the field, analysts at tactical operational centers, and translators operating from remote sites must work rapidly and accurately in triaging foreign language documents so that they or others in the shared workflow can prioritize which documents require additional analysis or *document exploitation* (DocEx). This section describes the experimental challenges and research results in assessing the effectiveness of Arabic-English MT engines for one document exploitation task, in enabling English speakers to identify essential elements of information in machine-translated documents that they otherwise would not understand.

5.3.2.2. Document Exploitation

When foreign language documents are collected by English speaking troops in the field, the time critical challenge is to determine the mission relevance of the information contained in those documents. In this context, “documents” refer to any of a wide variety

of items, including hard-copy materials such as newspapers, leaflets, official reports, receipts, supply lists or pocket litter, as well as soft-copy files with text, images, video, audio, or any combination of mixed media.

When documents are printed, handwritten, or spoken in languages other than English, the DocEx process will at some point include "content translation" in conjunction with other text-handling tasks that range from simple filtering or topic detection to binning or triaging by prioritized categories, up to more linguistically complex tasks of entity extraction, event identification, and summarization within and across documents (see Table 5.1). The most severe bottlenecks in the DocEx process in the field occur when massive quantities of foreign language documents need to be reviewed, and too few translators are available for this work.

Task	Description of Task
Publishing	Produce a technically correct document in fluent English
Gisting	Produce a summary of the document in English
Extraction	For documents of interest, capture specified key information in English: • *Deep Extraction:* Event identification (scenarios): id incident type & facts - task iii: Event Completion: id relations among elements, time, & place of event • *Intermediate Extraction:* Relationship id (ex. member-of, phone-number-for) - task ii: Wh-item extraction: id who-, where-, when-type elements of information • *Shallow Extraction:* Named entity recognition: isolate names of people, places, organizations, times
Triage	For documents determined to be of interest, rank by importance - task i: Topic id: bin document by topic
Detection	Find documents of interest
Filtering	Discard irrelevant documents

Table 5.1: Hierarchy of text-handling tasks proposed by Taylor & White (1998), augmented to include ARL's tasks i—iii

5.3.2.3. DocEx Tasks with MT

Shortly after Church (1993) described "good applications for crummy MT," ARL[8] decided to test a DocEx triage task as an application for FALCon[9], its laptop-based system for non-translators to scan in hardcopy documents, OCR the stored images to text, run the text through MT to convert it into "English", and then search the resulting text for English keywords (Fisher 1997; Fisher 1999). The goal was for document collectors in the field to convert found documents into "English" via FALCon, and then to triage the output text for mission relevance, reducing the translators' workload.

The FALCon systems, once in the field however, came into the hands of in-country translators; they valued the systems for quickly generating first-pass translations that they could readily revise and give to their English-speaking supervisor, who in turn would review and possibly post-edit their translations for inclusion in summary reports. Instead of *reducing the number of documents* in the translators' workflow by enabling others to triage documents, FALCon's utility came in *reducing the time-to-translate documents* in

[8] Army Research Laboratory

[9] FALCon stands for **F**orward **A**rea Language **Con**verter

the translators' workflow by speeding up their process of gisting documents. With evidence coming back to the lab for the utility of FALCon in the field, two types of evaluation questions were raised about providing support for these systems:

- *Developmental testing* (DT) conducted by the developers in the lab: how "good" are the MT engines? (measures of translation performance)
- *Operational testing (OT) conducted with potential users on working systems: how "effective" are users of the MT engines? (measures of task effectiveness)*

In 2000, we began conducting pilot studies to determine which MT engines conveyed sufficient information in their translations to support different levels of text-handling tasks. In 2004, we ran two large-scale experiments on Topic Binning and Wh-item Extraction and one large-scale pilot test on Event Completion (see tasks i-iii respectively, in Table 5.2) to assess three types of Arabic-English MT engines: MT-1 a rule-based machine translation (RBMT) engine, MT-2 a statistical machine translation (SMT) engine, and MT-3 a lexicon-based machine translation (LBMT) engine[10]. This section focuses on operational testing (OT) of MT engines for Wh-item Extraction (task ii), where the engines showed statistically significant differences on two response metrics. The section concludes with analyses of how closely these task-based (OT) results align with text-based (DT) automated metric scores on the same set of documents.

Who-type: people, roles, organizations, companies, groups of people, and the government of a country
When-type: dates, times, duration or frequency in time, including proper names for days and common nouns referring to time periods
Where-type: geographic regions, facilities, buildings, landmarks, spatial relations, distances, and paths

Table 5.2: Wh-types in Wh-item extraction (task ii)

5.3.2.4. Wh-item Extraction Task

On this task, the subjects were asked to read through a machine-translated text on their screen and "mark up" all word sequences that belonged to the assigned wh-type for that text (see Table 5.2). Each marked-up sequence appeared both highlighted directly in the text in the wh-type color code and copied as an entry in the screen's summary box with all wh-items extracted so far for the presented text[11].

The Arabic ground truth (GT) of wh-items was defined in the original Arabic documents. Then a corresponding English reference translation (RT) set of wh-items was derived with the translator of the task documents. Since subjects were asked to mark wh-

[10] Results were presented at TIDES PI meetings (2001, 2002) and NIST Open MT Workshops (2003, 2004). DARPA TIDES funded the initial task-based evaluation research; the Center for Advanced Study of Language at U. of Maryland funded the follow-on large-scale experiments. Tasks i and iii yielded no statistically significant differences in the accuracy of subjects' responses across MT engines: all MTs supported task i and no MTs supported task iii.
(Laoudi 2006)

[11] For further details, see Tate and Voss 2006 and Voss and Tate 2006

items in the MT output texts, a corresponding set of correct answers for scoring their responses also had to be identified in these texts. Two task developers and one adjudicator coded the answer set of wh-items by marking up each MT text alongside its corresponding RT text already marked with RT wh-items. Their answer set was called the *omniscient truth* set of wh-items, to reflect their use of the reference translations while marking the MT texts[12]. The task developers assigned a category code to each wh-item for computing inter-coder reliability and reconciling and recoding the texts. The codes of A (correct), B (almost correct), and S (split) are defined in Table 5.3. When no words in the MT output were clearly semantically or phonetically related to an RT wh-item, that item was coded Z for lost in translation. As an example, three different codings for the same underlying wh-item with different MT output results are shown in Figure 5.6.

A 1) Exact match, synonym, or paraphrase, where words are in grammatical word order 2) Contiguous phrase
B 1) Exact match, synonym, paraphrase, but wording out of grammatical order or 1') Partial match with some content loss, but wording in grammatical order 2) Contiguous phrase
S 1) Exact match, synonym, paraphrase, but wording out of grammatical order or 1') Partial match with some content loss, but wording in grammatical order 2') Non-contiguous phrase
Z Lost OR not recognizable

Table 5.3: Category codes for wh-items identified in MT output or lost in translation

GT: كتبت ريم الميع: في قصر بيان ...
RT: Reem Meeh wrote:
yesterday at Bayan Palace...
MT1: Reem wrote 'Lmeeaa: [S]
in a statement derelict,...
MT2: I wrote Rim almayai [الميع] : [A]
in the short statement,...
MT3: clerks move flowing :[Z]
in castle demonstration/statement?...

Figure 5.6: Example of the same wh-item underlined in parallel sentences: the Arabic GT, English RT, and three MT outputs with category codes in square brackets

To capture differences among MT engines' output and quality (independent of task results), we scored the individual wh-items by category code to measure translation accuracy and loss in the MT text proper. The category counts, accuracy rate, and loss rates are broken out in Table 5.4[13]. There were no statistically significant differences

[12] Note that the HTER post-editors are also ``omniscient'' in this sense: they read RTs to understand the SL text so that they can post-edit the MT output to convey its meaning exactly.

[13] Let $|A|$ stand for the # wh-items coded as A's. Precision was defined as $|A| / (|A| + |B| + |S|)$, Recall as $|A| / (|A| + |B| + |S| + |Z|)$, and Loss rate as $|Z| / (|A| + |B| + |S| + |Z|)$.

between SMT (MT-2) and LBMT (MT-3) in terms of available wh-items to extract (total A, B, and Ss), but RBMT (MT-1) showed a significant loss in wh-items (Zs only)[14].

	A	B	S	Z	Prec.	Recall	Loss
MT1 Totals	67	51	20	18	.49	.43	.12
Who	21	17	12	6	.42	.38	.11
Where	34	15	2	5	.67	.61	.09
When	12	19	6	7	.32	.27	.16
MT2 Totals	91	49	9	7	.61	.58	.05
Who	29	19	7	1	.53	.52	.02
Where	41	12	1	2	.76	.73	.04
When	21	18	1	4	.53	.48	.09
MT3 Totals	67	75	4	10	.46	.43	.06
Who	21	26	2	7	.43	.38	.13
Where	33	22	0	1	.60	.59	.02
When	13	27	2	2	.31	.30	.05

Table 5.4: Results of category coding of wh-items in MT outputs

We hypothesized that, if correct local word order were critical for wh-item extraction, then the LBMT (MT-3) would yield the weakest response rates. If word order however were not critical, then LBMT (MT-3) would outperform RBMT (MT-1) due to the latter's loss rate and match SMT (MT-2).

The results by MT engine show subjects did statistically significantly better on two of the three metrics: they found more wh-items (higher recall) and they were misled by fewer non-wh-items (higher precision) in the statistical MT-2 output than in the output of the other two engines. The results did not yield a clear ranking of MT-1 and MT-3: the lexicon-based MT-3 did a better job in support of higher hit rates, while rule-based MT-1 did better with lower false alarm rates. Curiously, the miss rates across engines do not differ significantly: subjects fail to detect roughly the same proportion of wh-items in the output of each engine.

The results by wh-type show subjects did statistically significantly better marking where- and who-items correctly than the when-items. They were least likely to miss the who-items. This contrasts with the MT results, where the miss rates were comparable across engines. Subjects were roughly equally likely to mark items incorrectly across wh-types. This contrasts with the MT results, where false alarm rates differed significantly.

To rank the MT systems on the basis of a single metric, we applied an approach from statistical decision theory: the three different performance rates - hit, miss, and false alarm rates - were combined with weighted cost estimates into a single loss function (Tate 2006). Based on several tests with hypothetical weights that we selected, the overall preference ranking of MT systems that persisted, where higher value (lower cost) is better than lower value (higher cost), was: MT2 > MT3 > MT1.

[14] MT-2 showed higher precision and recall than the other two engines (As only). Details are presented in (Vanni 2004).

5.3.2.5. Correlating Task-based & Text-based Scores

The next phase of the evaluation was to ask how well text-based (DT) scores on task ii documents track the task-based (OT) results on these documents. Four automated text-based metrics were run on all machine-translated documents, to capture the MT engines' differences on full sentences. These results, presented in Table 5.5, are consistent with the text-based precision and recall on wh-items presented in Table 5.4; MT-2 ranks above the other two engines. However, the automated metrics do not provide a consistent picture for ranking MT-1 and MT-3: BLEU and GTM rank MT-1 well above MT-3, while METEOR and TER rank MT-3 slightly higher than MT-1.

System	BLEU	GTM	METEOR	TER
MT-1	.088	.529	.385	.221
MT-2	.187	.617	.524	.370
MT-3	.055	.453	.397	.233

Table 5.5: Automated metric scores on task ii document sentences by MT engine

Corr	Level	System	BLEU	GTM	METEOR	TER
P	**system**		**.663**	**.468**	**.865**	**.86312**
S	document		.211	.193	.242	.231
S	document	MT 1	.140	.147	.109	.235
S	document	MT 2	.134	.303	.298	.111
S	document	MT 3	.298	.323	.311	.182
S	document	Who	.253	.292	.320	.198
S	document	Where	.229	.185	.252	.251
S	document	When	.250	.158	.237	.331

Table 5.6: Text-based automated metrics correlated with task-based Hit Rates, for Pearson (P) and Spearman (S) correlations, at system and document level

The standard for meta-evaluation of new automated MT metrics has, until recently, been to aggregate autometric scores and human subjective judgment scores to the system level, and then to interpret the strength of their correlation as a validation of the former in terms of the latter.

We note that high correlations between automated metrics and task ii hit rates can be achieved when results are aggregated at the system level, as shown in the top row of Table 5.6. This is most evident for METEOR and TER metrics.

The correlations drop significantly when the text-based and task-based scores are paired at the individual document level. The modification, shown in the second row of Table 5.6 also yields correlations that are much closer to each other and have a much smaller range in values for the different automated metrics.

Pursuing this result further, when document-level scores are paired within MT systems, the table shows monotonic and significantly positive associations between hit rates and automated metrics, for all four metrics. Similarly the table shows significant relationships after grouping the data within Wh-type[15].

[15] All table correlations are permutationally significant with p-values $<$ .01 (Tate 2008).

These correlation analyses support the initial conjecture for task ii: subjects perform less well when given documents translated poorly by a weak MT engine, while they perform better when given documents translated by a stronger MT engine.

5.3.2.6. Impact of Task-Based Approaches in MT Evaluation

Task-based approaches to MT evaluation seek to provide MT users with results that they can interpret relative to their own text-handling tasks. This section presented results of a large-scale DocEx experiment where subjects performed wh-item extraction on MT texts. The study found that the accuracy of their task responses yielded results consistent with a text-based measure of document translation-difficulty, derived from an automated MT evaluation metric. These results suggest that the task accuracy on wh-item extraction may, with further studies, be estimated from automated text-based metrics.

5.3.3. Using United States Government Language Proficiency Standards for MT Evaluation

Authors: Doug Jones, Wade Shen, Martha Herzog, Sabine Atwell, Hussny Ibrahim, and Dan Ding

5.3.3.1. Introduction

The purpose of this section is to discuss a method of measuring the degree to which the essential meaning of the original text is communicated in the MT output. We view this test to be a measurement of the fundamental goal of MT; that is, to convey information accurately from one language to another.

We conducted a series of experiments in which educated native readers of English responded to test questions about translated versions of texts originally written in Arabic and Chinese. We compared the results for those subjects using machine translations of the texts with those using professional reference translations. These comparisons serve as a baseline for determining the level of foreign language reading comprehension that can be achieved by a native English reader relying on machine translation technology. This also allows us to explore the relationship between the current, broadly accepted automatic measures of performance for machine translation and a test derived from the Defense Language Proficiency Test, which is used throughout the Defense Department for measuring foreign language proficiency. Our goal is to put MT system performance evaluation into terms that are meaningful to US government consumers of MT output.

5.3.3.2. Defense Language Proficiency Test

The official Defense Language Proficiency Test (DLPT) is constructed according to rigorous, well-established testing principles to measure the foreign language proficiency of military personnel trained by the Department of Defense and several other United States government entities. Developed and improved over the course of several decades, the DLPT is focused on the language proficiency standards recognized and used throughout the government.

In support of studies that preceded GALE to determine the accuracy vs. utility of MT, we constructed a variant of the DLPT, adhering to the test construct and design

principles, but making modifications to measure the quality of machine translations. This test became known as the DLPT – Standardized Translation Assessment with Reference, or DLPT-STAR.

Earlier DARPA work in MT evaluation incorporated (1) an informativeness measure, based on answers to comprehension questions; (2) fluency, a measure of output readability without reference to a Gold Standard; and (3) adequacy, an accuracy measure that did refer to a Gold Standard translation (White OConnell 1994). Later MT evaluation found fluency and adequacy to correlate well enough with automatic measures (BLEU). Since comprehension tests are relatively more expensive to develop, the informativeness test was not used in later MT evaluations, such as those performed by NIST from 2001-2006. In other work (Voss and Tate 2006), and as described in the previous section, task-based evaluation has been used for MT evaluation. These techniques measure human performance by exhaustively extracting `who', `when', and `where' type elements in MT output.

The DLPT-STAR also uses such factual questions, but this is not the only question type. The DLPT-STAR test items focus on the elements in the original text that are most characteristic of its proficiency level as defined by the Interagency Language Roundtable (ILR) Language Proficiency Scale. For example, Level 3 texts may have test questions about abstract linguistic formulations, hypotheses, or supported opinion essential to the meaning of the text. The test construct of the DLPT-STAR is based on the ILR scale, which is used throughout the United States government, so it provides Defense Department decision makers with results they can readily interpret. A description of a text classification scheme based on the ILR scale can be found in Child (1993). Some key points about the ILR scale for reading at Levels 1 through 3 are shown below:

Level 1 texts: contain short, discrete, simple sentences; generally pertain to the immediate time frame; often written in an orientational mode; require elementary level reading skill. For example, newspaper announcements are typically written at Level 1.

Level 2 texts: convey facts with the purpose of exchanging information; do not editorialize on the facts; often written in an instructive mode; require limited working proficiency. For example, newswire articles are often Level 2.

Level 3 texts: have denser syntax and highly analytic expressions; place greater conceptual demands on the reader; often written in an evaluative mode; may require the reader to `read between the lines'; require general professional proficiency. For example, newspaper opinion and editorial articles may be written at Level 3.

5.3.3.3. Experimental Design

The DLPT-STAR measures the quality of machine translation output in terms of readers' comprehension of the text as reflected by the accuracy of their responses to short-answer constructed-response questions. Subjects participating in the test were allowed five to eight hours to complete the experiments, depending on the specific test. Approximately 50 participants were recruited from MIT and the surrounding community for each test. All were native readers of English. The experimental tests were delivered

by computer. Subjects were shown each text and its associated questions simultaneously. They were allowed to work at their own pace. The interface of the delivery software allowed them to navigate between texts and take breaks as needed.

Each subject's responses were rated against a predetermined scoring protocol. The same scoring process, protocol, and standards were used for all responses, regardless of whether the subject used MT output or the reference translation (the control case). The raters were blind to the conditions of the responses they scored.

We expected that, if the reference translation were better than the MT output, the reference should yield a higher percentage of accurate responses than the MT.

5.3.3.4. Initial Study – 2004

The first DLPT-STAR test was based on authentic Arabic texts at ILR Levels 1, 2, and 3, with short-answer questions at the level of the texts. Texts were identified by a team with substantial experience in ILR text rating and DLPT development. The documents were presented in two conditions of English translation: (1) professionally translated into an appropriate English equivalent, and (2) machine translation output from state-of-the-art MT systems; this was often quite garbled.

This experiment showed that, when reading MT output, native readers of English could generally attain or surpass the passing score on Level 1 texts with 75% comprehension and on Level 2 texts with 76% comprehension[16]. However, with Level 3 texts, the result was 51% comprehension, well below the passing score. Comprehension of the Gold Standard reference translation was 95% at Level 1, 91% at Level 2, and 79% at Level 3, as shown in Figure 5.7. The results are presented in greater detail elsewhere (Jones *et al*. 2005).

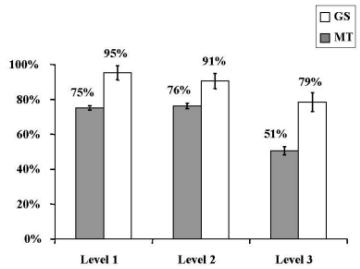


Figure 5.7: First DLPT-STAR Results

The lower results for Level 3 on the reference translation reflected the higher degree of difficulty of the original texts. Subjects may not have frequently read this kind of prose, even in their native language. However, we could not determine whether the somewhat weak performance at Level 1 on the MT output was due to systematic deficits in MT performance at Level 1 or to a mismatch between the texts and MT capabilities. Subsequent experiments in the GALE program were designed to address the ILR scale levels while using materials better suited for MT processing.

[16] In line with many language proficiency testing practices, we set 70% as a passing grade.

5.3.3.5. Second Test – Arabic MT

In 2006 during Phase 1 of GALE, we created a new variant of the DLPT-STAR, using materials specifically created to test the capabilities of the MT systems. We used the DARPA GALE 2006 evaluation data sets, used by several research sites for testing MT algorithms. We arbitrarily merged the MT output from three GALE performers.

The ILR levels of the documents ranged from Level 2 to Level 3; DARPA GALE 2006 data did not contain any texts that could be rated as Level 1. The absence of Level 1 documents was not accidental; the kinds of text associated with Level 1, such as newspaper announcements, advertisements, weather reports, etc., have not been part of the typical training or test data for MT evaluation and would not be of interest to the Department of Defense.

To partially compensate for the lack of authentic Level 1 material, we constructed questions about Level 1 elements (including personal and place names) found in Level 2 and 3 texts. A standard DLPT would use texts rated as Level 1, resulting in more variety in topic and testing points. It would also ask Level 1 questions only about Level 1 texts, Level 2 questions about Level 2 texts, etc. Our method of compensation is only partial, because it does not show the true performance of MT systems on Level 1 documents. It is possible that the performance we have seen at Level 1 may be atypical. Training of the systems on Level 1 texts would be needed to provide answers.

We selected approximately half of the DARPA GALE 2006 evaluation material for our test. There were twenty-four test documents, with balanced coverage across four genres: newswire, web data, broadcast news, and broadcast conversation. Our target was to have at least 2500 words for each genre; we exceeded this slightly with approximately 12,200 words in total for the test. We constructed an exhaustive set of questions for each document, approximately 200 questions in all. See Jones *et al*. (2007) for additional details.

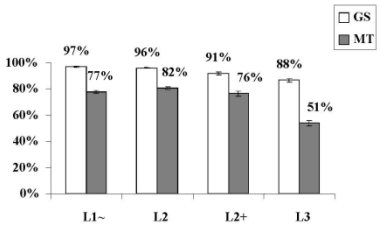


Figure 5.8: DLPT-STAR Results by ILR Level

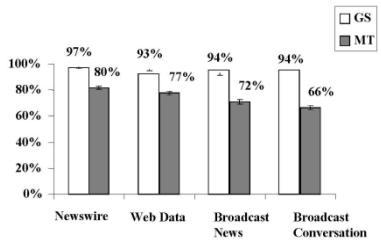


Figure 5.9: DLPT-STAR Results by Genre

The overall comprehension results for MT and for the Gold Standard (GS), professionally-produced reference translations, are shown by ILR level in Figure 5.8. The results by genre are shown in Figure 5.9.

The pattern of results for the second test using the DLPT-STAR was similar to that observed in the initial study:

- Level 1 performance was both lower than Level 2 and lower than expected.
- Level 3 was lower than Level 2, as expected.

Comprehension results for MT output were 77% for Level 1, 82% for Level 2, 76% for Level 2+, and 51% for Level 3. Gold Standard (GS) results ranged from 97% for Level 1 to 88% for Level 3.

Further analysis showed the low performance of MT at Level 1 to partially stem from the systems' incorrect translations of personal names, although these were essential to correctly answering many Level 1 questions. We observed that the name questions had 71% comprehension accuracy, compared with 83% comprehension for Level 1 items testing language features other than personal names. We should reiterate that MT systems have typically not had Level 1 documents as training data. Unfortunately, as long as this imbalance remains, the DLPT-STAR cannot be expected to show a consistent trend in performance across the ILR levels. Whether the addition of appropriate training data would affect performance is an open question.

Earlier (Jones *et al*. 2007), we also examined the relationship between comprehension rates and translation error. We compared comprehension rates with HTER, which counts the edits required for MT output to contain all, and only, the information present in a Gold Standard reference. A linear regression showed that subjects lose about 12% in comprehension for every 10% of translation error, with an R^2 value of 33%. The low correlation suggests that the comprehension results are measuring a somewhat independent aspect of MT quality. We feel this distinction is important. HTER does not directly address the fact that not all MT errors are equally important and that texts contain inherent redundancy that readers use to answer questions.

5.3.3.6. Using the DLPT-STAR for multiple MT systems in NIST's MT'08

Although we believed the overall design of previous DLPT-STAR tests was sound, we built the next test taking care to select documents across a broader range of topics and data sources. Also, previous tests compared machine translation output with professional reference translation output. The present test was designed to contrast different MT systems, using GALE phase 2 materials and administering the DLPT-STAR in the NIST 2008 OpenMT evaluation.

We selected ten Arabic-to-English MT systems and ten Chinese-to-English MT systems that were part of the NIST 2008 OpenMT evaluation run on the progress test set. We used all ten Chinese Systems that participated in the NIST evaluation. We chose ten Arabic systems, from the top, middle, and bottom of the range of BLEU scores attained in the NIST evaluation. In order to have a manageable number of conditions per test, we ran five MT systems and one GS reference translation with 100 subjects at a time.

We tested two of the GALE genres, structured text (Newswire) and unstructured text (Webdata). Overall, Arabic performance was higher than Chinese, and Newswire performance was higher than Webdata. The BLEU scores and the DLPT-STAR comprehension results are shown in Figures 5.10 and 5.11 for two of these cases.

Figure 5.10 shows Arabic Newswire results, the highest comprehension results. The systems are sorted according to DLPT-STAR comprehension results and are shown by the gray bars. The white bar at the right indicates performance on the reference

translations. The black bars indicate the BLEU score for the progress test set for each of the systems. System `MT-A7' has the lowest comprehension rate and the lowest BLEU score. However, the highest BLEU score belongs to `MT-A3', which is in the middle range of comprehension results. The highest scoring DLPT-STAR score is associated with a relatively low BLEU score. For comparison, Figure 5.11 shows the results for Chinese Newswire.

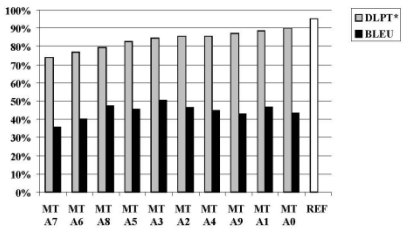


Figure 5.10: Arabic Newswire

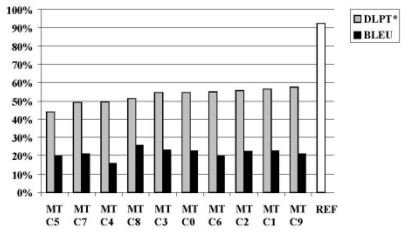


Figure 5.11: Chinese Web data

System-level correlation is shown in Figure 5.12. Each point indicates the overall BLEU score and overall DLPT-STAR score for one MT system in one condition. Since there are ten systems, two languages, and two genres, there are forty points in the cloud in Figure 5.12. At this broad level, the value is 74%.

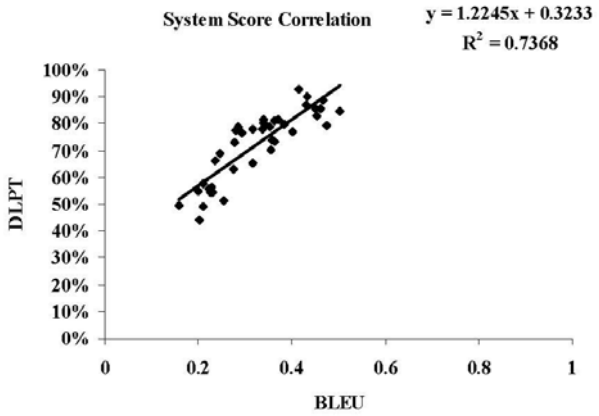


Figure 5.12: System level correlation

5.3.3.7. Impact of Language Proficiency Standards in MT Evaluation

Overall, we have found that Level 2 texts are a natural fit for successful machine translation. The MT systems seem to show an ability to convey concrete factual information that can be retrieved and used by the reader who does not need to understand the style, tone, or organizational pattern used by the writer.

Over the course of this series of experiments, we have observed increasing levels of comprehension demonstrated by readers of MT output. However, each experiment was run with a new DLPT-STAR test, making comparisons difficult. Now that we are able to

sequester the DLPT-STAR as part of the NIST OpenMT progress test set, we will have an opportunity to observe trends with the same test material.

5.3.4. Concluding Remarks on Task-based Evaluation

Both research studies discussed in this section report on human subjects performing language tasks using texts that were derived from the original foreign language source. The degree of distance from the original text depended on the quality of the translation, whether professional reference translation or machine translation output. The language tasks for the subjects were related to the communicative intent of the original text. That is, communication of the text's message was evaluated by requiring subjects to use all or portions of the message for a purpose. These two sets of studies approached the evaluation of communicative intent from different directions. Specifically, the first study focused on identifying the core factual information comprising the text to determine the relative accuracy of each system's output. The second study focused on determining whether subjects using machine translation output have adequate material to answer content questions about the most significant information in the original text. The first study examines the building blocks of a text, while the second examines the purpose for writing it. The interaction between subjects and text demonstrate the degree to which the original communicative intent is conveyed in the output. The subjects' accuracy in performing the tasks demonstrates the usefulness of the output. Most importantly, both studies move the research closer to evaluating real-world use of machine-translation systems.

Chapter 5.4 GALE Machine Translation Metrology: Definition, Implementation, and Calculation

5.4.1. Introduction

Authors: Mark Przybocki, Audrey Le, Gregory Sanders, Sébastien Bronsart, Stephanie Strassel, Meghan Glenn

One way to evaluate the quality of a machine translation is to note what edits (changes) a human editor will make to correct the meaning of the translation and to make the translation understandable (possibly also to make the translation reasonably fluent). We refer to that editor as a *post editor*. Ideally, the post editor will be bilingual, with a high-level understanding of both the source language and the target language (the language translated into), with the target language as his/her strongest language. However, post editing can be performed by a monolingual editor who compares the machine translation (MT) output to one or more carefully produced reference translation.

The GALE program uses the metric of *edit distance* to evaluate machine translation performance. As in many manual assessments of human language technologies, post editing can be implemented in a variety of ways. For GALE, the implementation of edit distance requires monolingual human post editors that refer to a carefully produced

reference translation, to make the minimum set of edits that completely corrects the meaning and makes the translation as understandable as the reference translation.

The TERCOM scoring software (Snover 2006) automatically compares the original (unedited) MT output to the post-edited version, and counts the number of changes. We refer to the number of such edits, divided by the number of words in the reference translation as the "Human-mediated Translation Error Rate", or HTER (Snover 2006).

This section will focus on HTER and post editing, where we will define HTER and describe its use in the GALE evaluation of MT. The entire post-editing process will be detailed, including the recruitment and training of editors, the software tool and guidelines used for facilitating the edits, the workflow management system and quality control (QC) methods employed, and how the output of the post-editing process is used for the final calculation of HTER. Where appropriate, this section will cover the interaction between the post-editing process and HTER scoring. GALE post editing is the largest manual scoring effort of MT metrology being employed in any of the current NIST coordinated MT technology evaluations.

This section also addresses the challenges of editor consistency, and how GALE evaluations use control documents to allow for year-to-year editor consistency measurement. It describes the year-to-year changes made to the post-editing process, including improvements to the editing guidelines, added flexibility to the post-editing software, and streamlining the post-editing process. We also include a section that describes how MT output is selected and assigned to particular editors, such that all GALE system translations are handled equivalently. Finally, we describe the overlap of work between particular editors and how this information can be used to identify consistent (consistently good or bad) editors.

Although human post editing of present day MT outputs is expensive, we have probably come some distance from the situation in the nineteen-sixties, when the [controversial] ALPAC report observed that,

> ". . . when after 8 years of work, the Georgetown University MT project tried to produce useful output in 1962, they had to resort to post editing. The post-edited translation took slightly longer to do and was more expensive than conventional human translation." (ALPAC 1966)

5.4.2. From TER to HTER

Authors: Mark Przybocki, Audrey Le, Gregory Sanders, Sébastien Bronsart, Stephanie Strassel, Meghan Glenn

We begin by explaining the adaptation of TER to a human-mediated evaluation measure, HTER. As described in Section 5.2.2.4, TER is defined as the measure of edit distance when editing the original MT output to exactly match a human reference translation. The scoring software simply compares the original MT output to the reference translation. But it is not necessary for MT system output to match a reference translation word-for-word in order to convey the complete meaning of the translation. Synonyms, phrasal reordering, and substituting pronouns or simplified forms of noun

phrases for already defined proper nouns are just a few examples of acceptable alternatives.

For example, suppose the reference translation reads, "*His black car had a dented bumper, and when the car was driven down the street, the observers called us to report that it was on the move*." In this case, there is no problem if the MT output reads, "*His black car had a dented bumper, and when the black car was driven down the street, the observers called us to report that the black car was on the move*."

Correspondingly, the GALE metric for MT evaluation is not a measure of the number of edits required to match the reference translation text TER, nor is it a measure of the number of word matches between the MT output and the sets of reference translations (as in BLEU or WER). Rather, it is a measure of the minimum number of edits required to match the meaning of the reference translation and make it equally understandable (HTER).

Deciding which textual alternatives alter the meaning or impair the understandability of the translation requires human judges. For the GALE evaluation of MT system performance, human editors compare the original MT output to the reference translation and manually modify the MT output using as few edits as they can manage to make the MT output contain the same meaning as the reference translation and be equally understandable. We refer to this process as *post editing*.

HTER (Human-mediated TER) is a measure of edit distance between the original MT output and the post-edited version of the original MT output. HTER is the official metric used to evaluate GALE MT system performance. Thus, where TER measures the edit distance to change the machine translation output into a human reference translation, in contrast HTER uses the same scoring software but measures the edit distance to change the same machine translation into the final post-edited version.

In evaluation, output from multiple MT systems processing the same input, are compared. Thus, it is necessary in the calculation of HTER to divide the number of edits for each system by the number of words in the reference translation rather than in the original system translation, so that the denominator is the same across each system. Using the token count from the reference as our denominator serves as a normalizing factor, allowing for direct comparison of HTER scores. Without normalization, systems could reduce their error scores by simply outputting superfluous text. For example, if a system translation was "They *traveled to Mexico*," an example of superfluous text might be "*The group undertook a trip and traveled to the country of Mexico*." Now consider in each case if the reference referred to a trip to Spain. In each example there would be one edit (substituting Spain in place of Mexico). The first example would have an HTER score of 25% error (1 error divided by 4 token words), while the second example would have an HTER score of 8% error (1 error divided by 12 token words).

The following sections explain the process of HTER calculation for the GALE program. The required infrastructure is reviewed, including the editing guidelines, the recruitment and training of editors, the editing software, and the workflow system required to manage the post-editing effort. We describe the methods employed to accommodate for editor inconsistency and fatigue which is a part of every human effort that imposes a heavy cognitive load.

5.4.3. Post-editing

Authors: Mark Przybocki, Audrey Le, Gregory Sanders, Sébastien Bronsart, Stephanie Strassel, Meghan Glenn

5.4.3.1. Post-editing Guidelines

The definition of HTER centers on what edits are to be made. The editors compare the MT output to a correct reference translation, and edit the MT output correspondingly. To date, the translations to which GALE has applied the HTER metric have all been translations into English (the target language) from two source languages, Arabic and Chinese. Thus, English was always the language being edited.

Four goals were discussed at length during the design of the editing guidelines. First and foremost is that the edits should completely correct the meaning. The second goal is that the editors should find the minimum set of edits that will completely correct the meaning. These first two goals amount to the underlying idea of the HTER metric. The third goal is that the edited translation needs to be understandable English, assuming the source material is understandable. This third goal is an important constraint on the editing process. A fourth goal that was discussed, and ultimately rejected, was to ask the editors to produce fluent English, if the source-language material was fluent.

5.4.3.1.1. The Process of Creating the Guidelines

The Post-editing Guidelines (NIST 2007) reflect the goals that were just mentioned. All parties to the GALE program agreed with the goal of completely correcting the meaning, but when various parties in the GALE community post edited several example translations, there were differences in what those various editors had corrected. As the parties discussed those differences, some decisions emerged as to the desired edits. NIST and the LDC created the first full versions of the post-editing guidelines in order to capture those decisions; the guidelines increase the level of agreement among editors regarding what constitutes a difference of meaning, and thus increase the level of agreement about the edits required to completely correct the meaning. The guidelines were refined through an iterative process of using successive versions of the guidelines to post edit example translations, looking at the edits that resulted, and modifying the guidelines appropriately. The guidelines continue to be reviewed after each phase of GALE evaluation, allowing post editor comments to be considered.

Through that iterative process, it became apparent that some editors were distinctly cleverer than others at finding ways to correct the text with fewer edits. Ways to correct the text with fewer edits were addressed at several places in the guidelines.

5.4.3.1.2. Displaying the HTER Value to Help Minimize the Number of Edits

Minimizing the number of edits was also addressed in the recruitment and training of the editors; seeking editors who would find it interesting to find a minimal set of edits. The main tool assisting editors to understand what constitutes fewer edits has been the post editing software itself, which was modified to display (see upper right window pan

in Figure 5.13) the HTER value for the segment currently being edited, an HTER value that resulted from the edits just made. This modification to the tool was effective because it made it easy for editors to try various alternative edits, with immediate feedback about their effect on the HTER value. In practice, having the editors consult the displayed HTER value for the segment has proven effective.

5.4.3.1.3. The Post Editor's Role

The job of the MT post editors may be summarized as follows: "Make the MT output have the correct meaning (the same meaning as the reference translation), using understandable English, in as few edits as possible."

The guidelines ask the editor to keep in mind that if the MT output adds/inserts information that is not present in the reference translation, those additions should be removed. Likewise, if the MT output omits information that is present in the reference translation, the missing information should be added.

Although the editors work on one segment at a time, the tool displays all the preceding (already edited) and following (not yet edited) segments in the current document. This allows the editor to see the complete context of the document, both as the reference translation and as the system output being edited. The guidelines ask the editor to keep in mind that the goal is to correct the meaning of the document as a whole, and that the MT systems are free to rearrange words and phrases over long distances (to/from other segments in the document). The editor is not to gratuitously alter those rearrangements if they do not affect the meaning. Without the display of context, judges could have difficulty identifying acceptable rearrangements. In practice, GALE data identifies the sentence segmentation for the systems, practically eliminating rearrangements.

Finally, we ask the editor to avoid wholesale copying of the reference.

5.4.3.1.4. A Condensed Version of the Post-editing Guidelines

- Make the MT output have the same meaning as the reference human translation: no more and no less.
- Make the MT output be as understandable as the reference. Similarly, try to make the MT output not be more or less ambiguous than the reference.
- Punctuation must be understandable, and sentence-like units must have sentence-ending punctuation and proper capitalization. Do not insert, delete, or change punctuation merely to follow traditional rules about what is "proper."
- Capture the meaning in as few edits as possible using understandable English. If words/phrases/punctuation in the MT output are completely acceptable, use them (unmodified) rather than substituting something new and different.

In case of conflicts among the four rules above, consider them to be ordered by importance. For example, never sacrifice the goal of correcting the meaning just to minimize the number of edits.

5.4.3.2. Post-editing Software

The Post-editing Software is a Java application that NIST designed and programmed. A strong collaboration between NIST and LDC allowed for frequent improvements of the software over time. Version 1.2.2 of the post-editing software was used for GALE evaluations at the time of this writing.

In order to ease the process of distributing the software to the annotators, a single JAR file bundles all of the required libraries. This includes the TERCOM (Snover 2006) library that computes the HTER score and provides a visual representation of the modifications made to the original MT output.

5.4.3.2.1. The MTF File Format

The Machine Translation File format (MTF) is the primary exchange format used throughout all GALE post-editing evaluations. MTF is based on XML: a DTD file defines the set of elements and attributes that may appear in a file, their respective relationships, the possible attribute values, and so forth. NIST chose to base MTF on XML for the following reasons:

- Given a DTD, any XML file can be validated using one of the many XML validating tools available.
- XML handles Unicode encodings.
- XML syntax is fairly simple and allows for easy manual editing of the contents.

The root element of an MTF file is a container for one or more documents, which in turn contain one or more segments. Usually, one segment corresponds to a single sentence. Several attributes are used to identify the data contained in the file, allowing tools to automatically parse the contents of a set of MTF files.

Because members of each editing team work successively on the same file (see Sections 5.4.3.3.2 and 5.4.3.3.4), the MTF format embeds facilities for the team members to communicate. One such feature, the *Request For Review* attribute, is used when an annotator decides to notify a supervisor of any inconsistency in the guidelines or, more realistically, in the data being edited. For example, an annotator may want to report that a reference sentence requires a review if he/she thinks there might be an error in the reference translation.

5.4.3.2.2. The Post-Editing Interface

The main window (see Figure 5.13) is divided into three main panels. The leftmost panel contains the reference translation for a given document. The middle panel is where the translation of the document is edited. The rightmost panel is used to display the HTER representation (as described in Section 5.4.3.1.2) and the original version of the current segment.

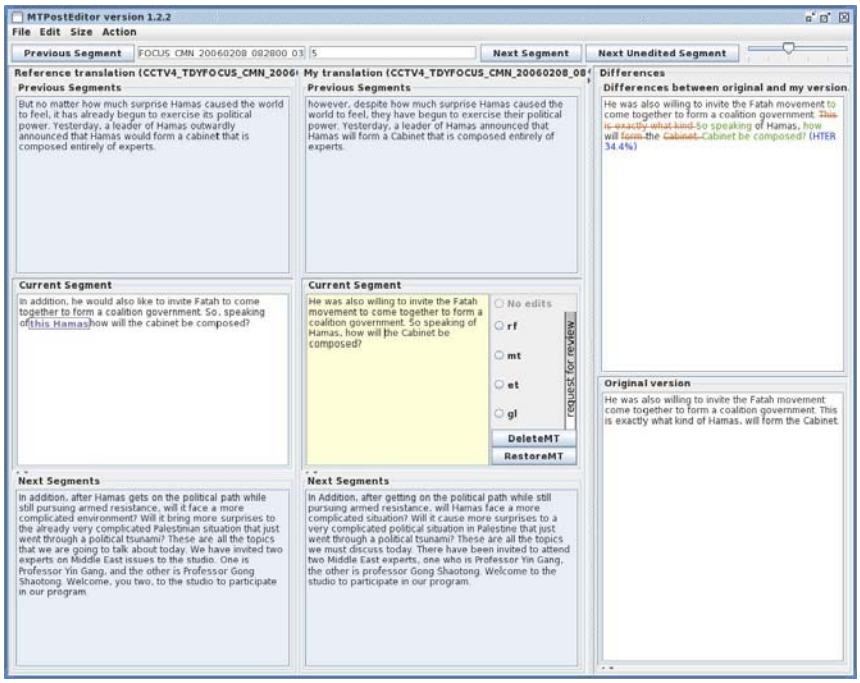


Figure 5.13: Main window of the post-editing tool

This leftmost panel (with the reference translation) is divided into three text boxes: the top box displays all previous segments for this document, the middle box displays the current segment, and the bottom box displays all succeeding segments for the document. Some reference words, or groups of words, may have associated alternate translations. If so, these words are highlighted and alternate translations are shown via a drop-down list that is invoked by hovering the mouse pointer over the highlighted text.

The middle panel (with the edited translation of the document) is also divided into three text boxes, containing respectively all the previous segments, the segment currently being edited, and all the succeeding segments yet to be edited. Additional buttons are provided to allow the annotator to flag the current segment, in case it cannot be edited directly.

5.4.3.3. The Evaluation

5.4.3.3.1. Recruitment and Training of Post Editors

Linguistic Data Consortium recruits Philadelphia-area native English speakers with training in copy-editing, proofreading, creative writing, or journalism. First-time applicants residing outside of the region are excluded because the training process requires a visit to LDC in Philadelphia. All candidates receive by email a pre-test "kit," containing a brief set of instructions, the annotation tool and the test file to edit. The test file consists of eleven sentences which LDC selected from GALE Phase 1 machine translations that represent the two source languages and multiple genres and machine

translation systems. The eleven sentences were also selected to represent a range of editing difficulty, so that the first few sentences of MT output require fewer changes, and the last sentences require more - and possibly more creative - edits.

Managers score and review the test kits carefully, looking for a basic post-editing aptitude in the test kit responses: primarily, correct spelling, incorporation of the full meaning of the Gold Standard reference in the test edits, and no extraneous information. After examining the pre-test results and eliminating outlier candidates, LDC invites qualified applicants to LDC's offices for an intensive group training session.

The training session focuses on the program goals and the MT post-editing guidelines developed jointly by NIST and LDC. Managers display a set of editing examples to demonstrate editing possibilities and pitfalls. Following the training session, applicants re-edit the test kits so that managers can observe what they would do differently after learning more about the task. Those who continue to produce edits that convey the same meaning as the Gold Standard translation and who make only necessary changes to the MT are selected for the project.

Many editors return for multiple cycles of post-editing within the GALE program. Returning editors are required to attend a re-training session at LDC. The re-training session focuses primarily on the same points as the original training session, but managers provide more time for questions and examples. In addition, the group works through multiple examples together. Editors are also encouraged to come to LDC for "office hours," or regular opportunities during the course of a project to sit down with a manager to discuss editing questions in general or review specific kits in detail.

Before beginning work on GALE evaluation data, editors read the guidelines carefully and complete a starter kit, a set of documents that reflect attributes of the data for the current phase. The starter kit reinforces their knowledge of the post-editing rules and allows editors and LDC staff to solve technical problems. It also offers managers another opportunity to evaluate editors and to answer many task-related and procedural questions before the project begins in earnest.

5.4.3.3.2. LDC Workflow and Quality Control

Editors work remotely, accessing post-editing assignments through a web-based workflow management site, which LDC developed for this task. The workflow system assigns kits, or files containing approximately 1200 words of translated material, to editors. Editors are assigned the role of first pass or second pass editor at the beginning of the project. The primary role of a first pass editor is to perform the initial editing of the MT output, finding the minimal number of edits required to capture the complete meaning of the reference translation. The primary role of the second pass editor is to review and check the work of the first pass editor. The second pass editor determines if the meaning of the edited MT output matches the reference. If correct, the editing is left alone. The secondary role of the second pass editor is to determine if appropriate modifications can be made in order to reduce the HTER score. Second pass editors are paired with first pass editors, to form a team. In most cases, two editors will be paired for the duration of a project. Translations are reviewed by two independent editor teams of first- and second-pass editors.

The workflow system serves both the editors and the LDC managers. LDC managers are able to view and manage users and assignments, as well as, backup the project through the system. Assignments, the kits, for each editor team are loaded into the workflow system at the start of the project. Editors check out kits from this system, working with one kit at a time. After a first pass editor checks in a completed kit and no problems are identified by an automated scoring process, the kit is automatically assigned to the second pass editor.

The workflow system was designed as a central information resource for editors. Here they can view their file assignment lists, verify expected payment per kit, and find links to other project resources. Editors are also able to check the status of kits in their queue, and monitor their first- or second-pass partner's progress. Second-pass editors benefit from a "nudge" option in the workflow system, which allows them to send a polite reminder to their editing partner that they are waiting for data.

LDC quality control measures are in place to catch careless errors or alert managers to potential problems during the editing process. In addition to managing file assignments, the workflow system supports such quality control measures. When an editor checks in a file, the workflow system triggers various automatic and manual LDC quality control mechanisms. For example, the first kit submission of every editor is flagged and held in a separate queue until approved by a manager. Scripts automatically score and check incoming kits to identify potentially problematic kits; these include kits with high HTER scores or with unedited segments. LDC managers also spot-check kits daily, and provide feedback to editors accordingly, in order to satisfy the quality requirements for the project.

5.4.3.3.3. Reducing the Cognitive Burden

As with most (possibly all) manual evaluations of translation quality, the difficulty of the task presented to the human judges has a direct bearing on the quality or consistency of their effort. This section describes specific efforts that were made to reduce the cognitive burden placed on the editors.

As was mentioned in the previous Section, data units or "kits" that are to be edited by the editors are kept to a reasonable size. For the first GALE evaluation of MT performance each kit contained approximately 1500 words of translated text to be edited. Post-evaluation editor comments led to reducing the kit size to approximately 1200 words. While post editing is performed remotely (left to the editor's choice) and the editors are under no scrutiny to work through a kit absent lengthy breaks, we have estimated that an editor spends approximate two to four hours editing each kit. The goal was to be sure kits were small enough to finish one kit per day. Kit size was a primary consideration in defining which documents were grouped into a specific kit.

A second consideration in the kit definition process was the original source language. We have observed through automatic and manual measurements of MT quality that the English translations of Arabic source data tend to be better than those from Chinese source data. With that in mind, kits were created as to mix, and preferably alternate between, documents that originated in the two languages. The thought being that the lower MT quality would require more thought, more careful editing, to achieve the post editing goal. This second consideration was not always possible. For example in the first

MT evaluation during phase 3, only one team participated and only the Arabic source data was processed.

The third consideration designed to reduce the cognitive burden on editors related to mixing the genres in each kit. To the extent possible NIST sought to mix audio and text sources and within that grouping NIST sought to interleave translations from each of the four genres (text from newswire and web; audio from broadcast news and broadcast conversations). Again, through automatic and manual evaluations of MT performance by genre, it has been observed that system performance was generally better for the structured sources when compared to the unstructured sources (translations of newswire data were generally better than translations of web data, for both languages; and translations of broadcast news data were generally better than translations of broadcast conversations, for both languages).

And finally, as was mentioned in Section 5.4.3.1.4, a one-page reminder of the post editing goals was produced as a quick reference guide. The full post editing guidelines are too comprehensive to expect editor memorization. This one-page reference guide emphasized the order of priorities and general philosophy behind the editing process and served new editors as a tool that could quickly be committed to memory.

5.4.3.3.4. Post-editing Protocols - Teams: first pass editors and second pass reviewers

This section describes the post-editing protocol used in the GALE program and the changes that were made in later phases of GALE such as to combine the first pass editors and second pass reviewers into a team, streamlining the post-editing process.

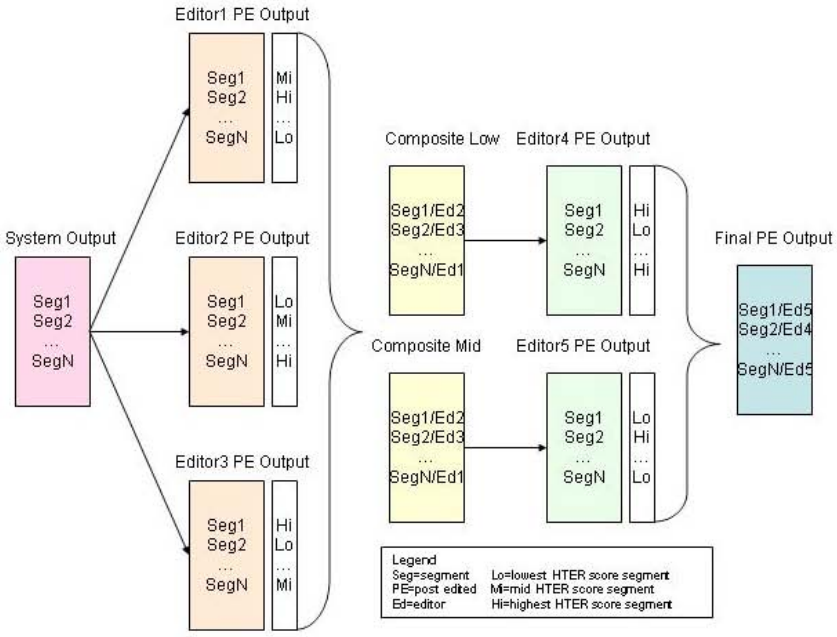


Figure 5.14: Diagram outlining the post-editing protocol used in GALE phase 1

We designed the GALE post-editing protocol to maximize repeatability and improve validity by employing a multi-stage process. Figure 5.14 summarizes the protocol used in the first year of GALE. In the first year of GALE , the editing process involved three truly separate stages: first-pass editing, second-pass reviewing, and a final stage in which a composite document was assembled and scored. In the first stage, three independent first-pass editors post edited each system's output translation. The result was three first-pass post-edited versions of the same document. The HTER score was computed for each segment of each of those three versions.

In the second stage, the two composite outputs were reviewed by different second pass editors who acted as a reviewer and made additional (or different) edits and corrections if needed. The focus of the reviewer was ensuring that the meaning of the post-edited translation matched the meaning of the reference translation. Usually the editors who served as reviewers were more experienced editors. The second pass editors were primarily concerned with ensuring the accuracy of the edits. This stage resulted in two second-pass post-edited versions. The HTER score was computed for each segment of each of the two second-pass versions.

In the third stage, a final composite output was created from the two second-pass post-edited versions of each segment by choosing the version with the lower HTER score for each segment. The final HTER value is computed over that final composite output.

We turn now to elaborating on the rationale for this process. For HTER to be meaningful, the post-edited output must be consistent and accurate. Accurate is defined as (1) completely correcting the meaning, (2) being as understandable as the reference, and (3) using the minimum number of edits necessary to accomplish those two goals. Consistent is defined as repeatable, which in practice means different editors would make similar edits on any given system output. However, the post-editing task is highly subjective despite the fact that editors have gone through a rigorous training process. Thus, multiple editors were used to post edit each system output, in an attempt to ensure editing consistency.

In the first stage, for practicality reasons (time and cost) we chose the number of first-pass editors to be three although a higher number would have been more desirable. As has been mentioned, the post-edited translation should convey the same meaning as the reference translation. This does not mean the post-edited translation should have exactly the same wording as the reference translation. Using three independent first-pass editors generated a wider range of alternative ways to edit the translation so as to minimize HTER. The second-pass review editors ensured that the meaning was not compromised. The reviewer is necessary because minimizing HTER conflicts with improving the meaning of the translation.

Making as few edits as possible during the editing process is important to a meaningful HTER score (recall that the idea of HTER is that it is a measure of the minimum edits required to correct the MT output). Without keeping an eye on the number of edits being made, the HTER score could be falsely inflated. Finding the minimal number of edits is, however, an ideal; there is no benchmark to compare to. And the task is difficult, with clear differences among editors in their ability to find a minimal

set of edits (as discussed at the end of Sections 5.4.3.1.1 and 5.4.3.1.2). Therefore, to increase repeatability we create the composite version using the segments with the lowest HTER scores from the available second-pass post-edited outputs. In this context lower HTER scores are assumed to be better editing.

In the current phase of GALE, and for future phases, the post-editing protocol employs a streamlined two-stage process: in the first stage each system output is post edited by two independent teams of editors thus creating two post-edited versions, and then in the second stage (exactly as in the previous third stage) we create and score a composite version.

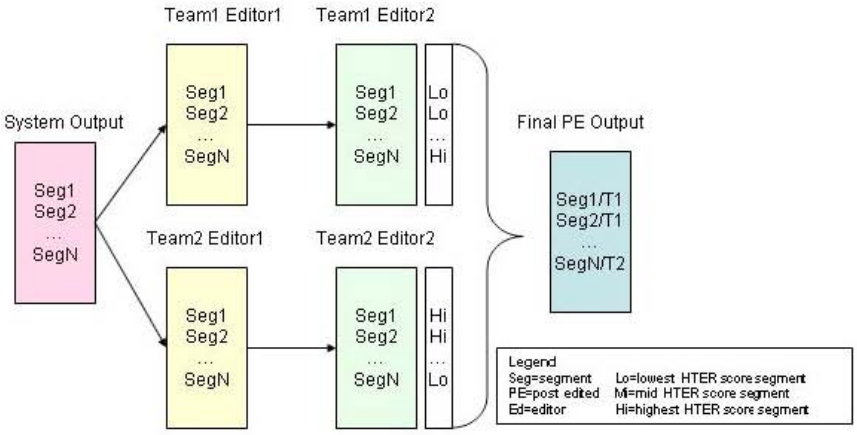


Figure 5.15: Diagram outlining the current post-editing protocol used in GALE

Each post editing team consists of two editors, one acting as the first-pass editor and the other serving as the second-pass reviewer checking the edits of the first pass editor. Together, the editors of a team produce one post-edited version. Thus, the two teams produce two post-edited versions. Unsurprisingly, with the variability that exists in human language, the resulting edits between two teams may at times vary, sometimes drastically. Ideally, five or more teams would be used for post editing each document and the differences between teams could be averaged for a better understanding of HTER scores. The practice of limiting to two teams has been dictated by both cost and the amount of data post edited for evaluation. HTER is calculated for each segment of each of the two versions. We select the version of each segment with the lower HTER, combine them into a composite document, and score the composite document. Figure 5.15 summarizes the protocol used in GALE at the time of this writing.

5.4.3.3.5. Potential Problems and Attempted Solutions

There are two identified weakness in using the described post-editing protocols for MT evaluation over successive years, and both have to do with the human aspect of the task.

Despite LDC's best efforts, it is not possible to retain the same set of post editors for duration of a five year program. During each phase of GALE, new editors are added to

the team and some choose not to return. The quality of each post editor can range from those who are exceptional at solving the puzzle (finding the least number of edits required to modify MT output to contain the exact meaning of the reference translation using equally understandable English) to those who solve the puzzle in less than the optimal fashion. Some editors may be aggressive in the amount of editing they perform to capture the meaning while others may be too lenient, taking too much editor liberty in deciding the meaning is equivalent. It is through the LDC's quality control process of spot checking editor performance that such trends may found. LDC also compares each editing team's HTER scores to those of the alternative version. When a team is found to be producing consistently lower or higher HTER scores than the corresponding editing team, a flag is raised and a review of their work is conducted. On occasion, an editor has been counseled due to poor editing skills, but one has never been let go during the evaluation.

As the pool of editors change from evaluation to evaluation, "control documents" are used to measure the editing consistency across evaluations. This is accomplished by inserting original translations that were previously edited, in the preceding evaluation, into the current editing task. When the documents are edited by different individuals, this process allows the measurement of the average difference on a subset of documents between two evaluations. Major differences are not expected, but can result from changes in editing protocols or from changes in the performance of the editors. Major differences prompt investigation. Editor differences between two evaluations can help explain system performance changes between the two evaluations. If editors are found to be particularly picky one year, they can unknowingly mask system improvements.

5.4.3.4. Conclusion

This section described the evaluation metrology used for the evaluation of MT performance for the GALE program. HTER as defined is the primary metric for GALE evaluations. NIST and LDC jointly created the post editing guidelines. NIST has created the evaluation datasets. The Linguistic Data Consortium has coordinated and supervised the post editing work for the GALE evaluations.

5.4.4. Machine Translation Assessment

Authors: Mark Przybocki, Audrey Le, Stephanie Strassel, Calandra Tate Moore, and Gregor Leusch

5.4.4.1. Introduction

The DARPA GALE MT evaluations test a system's capability to accurately and fluently produce an English translation for language data that originated in one of two languages, either Arabic or Chinese. The official evaluation metric is HTER as described earlier in Sections 5.2.3 and 5.4. The evaluation paradigm requires local implementation, that is, GALE teams receive the evaluation source data, allowing processing to take place locally using their distributed translation system network. Due to the complexity and the costliness of post editing, each GALE Team submits a single system for evaluation representing the one-best translation. Contrastive systems and system components are not

evaluated. In essence it is the quality of the returned hypothesized translations that is being evaluated.

There are two tasks evaluated in each GALE MT evaluation. These tasks differ only by the medium of source data being processed. We refer to the two tasks as the *Translation task*, when the data medium is text, and the *Transcription task*, when the data medium is speech or audio files.

5.4.4.2. Translation Task

The GALE MT *Translation task* requires systems to accurately and fluently translate foreign language text data into English. The text data has been drawn from various online news and web forums or blog sources. The evaluation data sets were designed to include some source providers that were represented in the allowable training data and some source providers who were not. The data encoding for both input and output files has remained UNICODE UTF-8 encoded data, throughout each phase of GALE.

Although the definition of the *Translation task* has enjoyed relative stability throughout each phase of GALE, there were two noticeable changes that occurred between Phases 1 and 2. The first allowed systems the use ground truth sentence segmentation, as defined from the original source data. In review of the phase 1 evaluation protocols, it was noticed that too many post-editing errors may have been improperly influenced by the automatic alignment process (Matusov 2005) that was used to align the system hypotheses to the reference translation. An automatic alignment process was necessary since the segmentation used for each team's system translations were created independently of each other and independently of the reference translation segmentation. Although post editors are trained to consider information in a document as a whole, allowing information to exist outside the current segment under focus, in practice many post editors were inconsistent with how they handled information in adjacent segments which could lead to over editing. By providing the true segmentation marks for the following phases these types of errors were and will be eliminated. The second modification was to reduce the size of the evaluated document. Rather than evaluating the contents of each full document, which ranged in size from about a hundred words to several hundred words, smaller cohesive subsets of the larger document were defined with each targeted to have about 200 words, thus allowing for more data points during analysis. This change was mainly due to a change in the program's targets. For Phase 1, the target was average accuracy. For the following phases, the target was set as the percentage of documents that exceeded a minimum accuracy target. To get a proper value for the percentage of documents, we increased the number of documents evaluated.

5.4.4.3. Transcription Task

The GALE MT *Transcription task* requires systems to accurately and fluently produce English transcripts from foreign language audio sources. Like the *Transcription task*, the evaluation data sets for Transcription were designed to include some source providers that were represented in the allowable training data and some source providers who were not. The source language audio files for the *Transcription task* were provided to the GALE teams as NIST à SPHERE (Garofolo 1994) formatted waveforms, most

likely requiring systems to use an automatic speech recognizer (ASR) as the front end ideally optimized for use with a translation system. While both the ASR and MT system components could be measured separately the focus of the GALE evaluations is not on the individual component performance but rather to evaluate the system performance as a whole.

As with the before mentioned *Translation task*, the *Transcription task* too had similar changes to begin allowing systems to use segmentation information, but this change for the *Transcription task* occurred after Phases 2 and was motivated out of the necessity to save time and to reduce the size of the evaluated audio file rather than any type of observed errors.

5.4.4.4. Data

Each succeeding GALE evaluation phase requires a new test set to be created for both evaluation source languages (Arabic and Chinese). The reason for the requirement for new test material arose because the performers used the results of previous evaluations for error analysis. This section defines the evaluation test sets used in the GALE phases. The evaluation test sets have been collected by the Linguistic Data Consortium (LDC). The actual selection of the data files to be used in evaluation has been a combined effort between NIST and the LDC, with final approval coming from DARPA.

The National Virtual Translation Center (NVTC) was responsible for generating the Gold Standard reference translations for the first evaluation (Phase 1) and the LDC has been responsible ever since.

Genres

At the highest level, the GALE evaluation data can be categorized into two distinct categories, structured data and unstructured data.

Structured data is defined as that which is cleaner, well prepared and most likely has been reviewed and edited before being published. There are two genres that represent structured data. The first is *newswire*. Newswire is text data that is typically a single coherent news story, examples can be viewed from any online news reporting agency. For the first phase, 3 Arabic and 2 Chinese sources were used in the evaluation. There was a concentrated effort to include more variety of source providers in the following phases (7 Arabic and 6 Chinese sources).

The second genre of structured data is *broadcast news* which contains smaller news clips extracted from a larger audio broadcast. As with text, broadcast news data represents clean audio broadcasts, usually scripted and this data has probably gone through multiple reviews before being broadcast. This data is typically void of conversational and spontaneous data. As with newswire, the first phase of GALE used 4 Arabic and 4 Chinese sources and this set was augmented to 17 Arabic and 14 Chinese sources.

Unstructured data genres, as the name implies, carry a more informal tone. In text data, the informality is found in the form of web data, blogs and forums where keyboarders are more likely to write in fragments, with typing errors, and using

unconventional phrases and abbreviations. We refer to the unstructured text genre simply as *web data*. For audio data the informality comes in the form of call-in talk shows and live reports from the field, including unscripted interviews. This genre is referred to as *broadcast conversations*. For Phase 1 UT, only 2 Arabic sources and one Chinese source of Newsgroups were used and for UA, 6 Arabic and 3 Chinese sources. This was augmented in following phases to 7 Arabic Newsgroups, 12 Weblogs, 2 Chinese Newsgroups and 9 Weblogs for UT. For UA, 24 Arabic sources and 14 Chinese sources were used in later phases.

We will use the following abbreviations throughout this chapter: NW to refer to the newswire genre, WB to refer to web data, BN for broadcast news, and BC for broadcast conversations.

Test Set Size

A reoccurring issue in defining an evaluation data set is to answer the question of "how much data is required for the test?" For GALE the "how much data" must be taken in consideration with the practical question of "how much will it cost" for the post editing of resulting system translations.

In Phase 1 the evaluation test set size was measured via source word counts with the goal to have approximately 10,000 source words represented in each of the four genres, for each source language. While this may seem relatively straight forward, a couple of assumptions were made. First, for Chinese text data, 1.5 native Chinese characters were estimated to correspond to one word. And since the selection of evaluation data was from audio (before reference transcriptions were available) an estimation of the rate of speech in Arabic and Chinese BN and BC data was required. The rate of speech estimates used for the various audio data sets are listed in Table 5.7 The adjustment between Phases 1 and 2 was made after post evaluation analysis identified that much more data was post edited in phase 1 than anticipated.

Language/Genre	Estimated Rate of Speech words per hour	
	Phase 1	Later Phases
Arabic BN	6000	5700
Arabic BC	6000	5700
Chinese BN	10,000	11,000
Chinese BC	10,000	11,000

Table 5.7: Estimated rates of speech for GALE audio evaluation data based on calculations from lower quality training data

Following Phase 1, the evaluation test set size was increased by 50%, that is, the goal was for each genre to be represented with 15,000 source words (for each language) and the number of documents was increased as explained above. Table 5.8 lists the actual words per genre as measured by the source and reference word counts. The decision to limit documents to smaller snippets (cohesive topic segments) allowed for the inclusion of many more document samples.

Language/Genre	Phase 1				Later Phases			
	SRC Words	REF words	Doc Count	AVG Doc length	SRC words	REF words	Doc count	AVG Doc length
Arabic BN	12k	15.5k	24	636	14k	20k	69	218
Arabic BC	11k	15k	13	1130	14k	20k	60	250
Arabic NW	10k	14k	51	182	14k	18k	68	220
Arabic WB	10k	12.5k	29	309	14k	19k	69	219
Chinese BN	13k	14.5k	18	777	16k	17k	70	215
Chinese BC	14k	13k	11	1126	15k	15k	53	283
Chinese NW	10k	10.5k	36	240	16k	17k	74	201
Chinese WB	10k	10.3k	19	485	15k	17k	66	227

Table 5.8: Word counts for phase 1 and phase 2 evaluation test sets. Source words are counts of native word tokens and reference words are counts of the words in the English translation.

The fluctuation in test set difficulty from year-to-year has been an issue that NIST has struggled with in many human language technology evaluations. In the DARPA EARS program, NIST began making use of a "progress" test set that was treated with special instructions so that it could be reused in subsequent evaluations, allowing system improvements to be better tracked over time. This scenario only works if the evaluation data can remain completely blind from the systems and participants. In GALE evaluations this is not possible. The evaluation data is examined in excruciating detail post evaluation. Post editors completely probe the data, and participants are encouraged to perform error analysis in order to improve their systems. Given these constraints, the program explored two alternatives for dealing with year-to-year test set difficulty differences. The first involved controlling the evaluation test set selection process, and the second made use of a few control documents allowing year-to-year test set calibration. These two ideas are further explained below.

Defining the Evaluation Epoch

Test set creation begins with selection of the evaluation (and optionally, a corresponding development test) epoch. Several factors are considered when choosing the evaluation epoch. First, the epoch should occur during a period of relative stability for LDC's ongoing data collection. For instance, LDC's broadcast collection system is taken offline once or twice a year to perform software upgrades and ongoing system maintenance, and an evaluation epoch that coincides with one of these maintenance windows would be subject to recording gaps. Similarly, programming schedules on the broadcast networks that LDC subscribes to are subject to variation during the time change between standard to daylight savings time, and so recording schedules may be unreliable for the few days surrounding this transition. Programming schedule disruptions are also common during holiday periods, for instance Chinese New Year and Ramadan. A second consideration in selecting the evaluation epoch is topical content in the news. While GALE evaluation data is intended to be representative of the linguistic challenges that exist in real world data, it happens occasionally that a single topic will dominate news coverage for days or weeks on end, and care must be taken to avoid selecting an epoch in which one single topic will predominate. For instance, the earthquake that

occurred in the Sichuan region of China in May 2008 completely dominated Chinese news reports for several days on end, disrupting normal programming considerably.

Consideration must also be given to what data sets have already been "exposed" as training data (within GALE, or as part of other technology evaluation programs that GALE sites may have been part of); evaluation data should be blind and so previously-exposed epochs and possibly sources are avoided. Finally, an evaluation epoch must be selected from early enough in the current phase to ensure adequate time to prepare Gold Standard references before system evaluation takes place.

At minimum, GALE evaluation epochs are one month in duration. This is necessary to ensure sufficient data volume and variety to support targeted selection. If a development test (devtest) set is to be defined, it typically comes from a time period directly preceding the evaluation epoch. Typically LDC and NIST collaborate to define an appropriate evaluation epoch, and the plan is endorsed or refined by the GALE data committee.

Evaluation Candidate Pool and Manual Selection

Once an evaluation epoch has been defined, LDC creates the evaluation candidate pool. The candidate pool is very large, typically hundreds of times larger than the final evaluation set. It consists of source data for all four genres that were collected during the evaluation epoch, processed to conform to the standard GALE source data format, and (where applicable) subject to manual or automatic auditing for quality and content (Part 1 in this book).

The candidate pool serves as input to the first stage of data selection. LDC annotators review the contents of the candidate pool to identify portions of documents or files that meet specific selection criteria. LDC has developed a customized user interface for doing data selection. (See Figure 5.16.) The interface uses a database backend to track annotation decisions. The process begins by dividing the candidate pool into randomized file lists. Each list includes up to 200 files, drawn from a single language and genre and either mixed or uniform with respect to sources/programs and epochs, depending on requirements. All files in the candidate pool appear on some list. At the start of a work session the annotator logs into LDC's Annotation Workflow System (AWS) ((Maeda 2009) and Part 1 in this book) and chooses Eval Data Selection from their assigned list of tasks. AWS then launches the data selection toolkit and pre-loads the next unassigned file list from the pool of available data. After the tool launches, the annotator sees the list of candidates to review, along with the first line of text in the file (where available) and the file size (in tokens or duration). The annotator clicks on each filename in turn, whereupon the tool displays the full text and/or audio for that file. Newswire candidates are presented for manual review as individual story units, with story boundaries pre-defined by the source data provider. Web text candidates are presented for review as threads, which consist of a single entry or post on a given topic, plus all the follow-up entries, posts, messages and/or comments responding to the original post. Broadcast candidates are presented for review as single audio files that correspond to a recording of one episode of a single program, typically 20, 30, or 60 minutes in duration, plus the corresponding automatic speech recognition (ASR) output for that recording (Part 1 in this book).

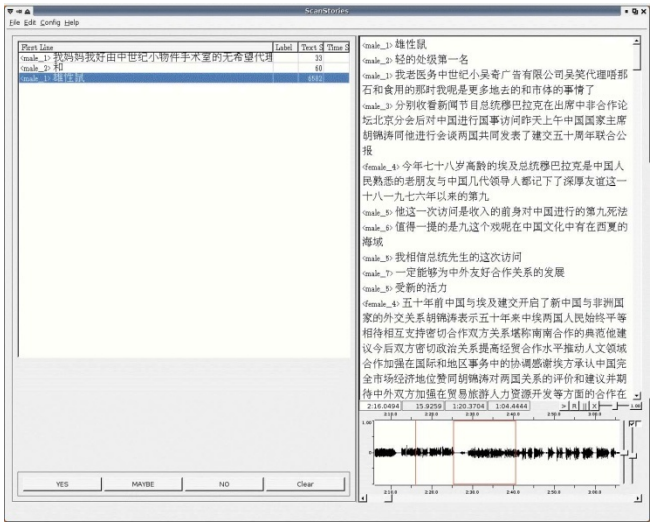


Figure 5.16: Audio and Corresponding Text Display for Selected Story

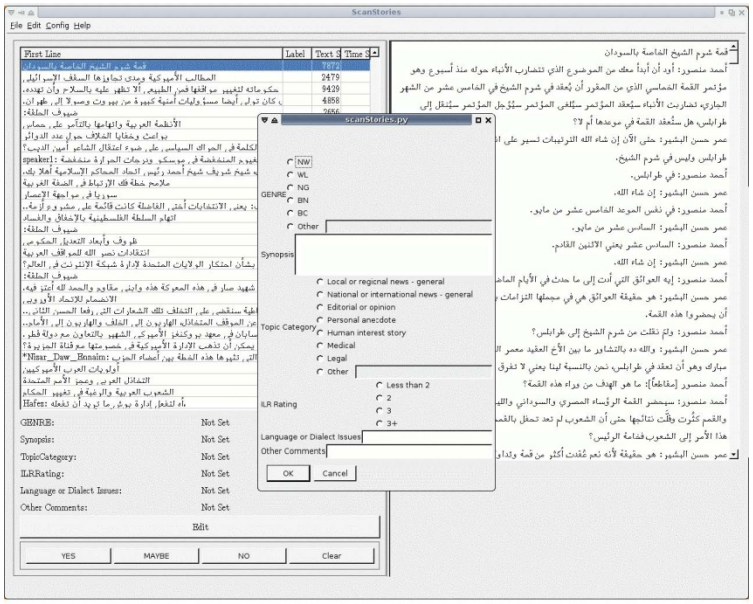


Figure 5.17: Completing Judgments for Selected Story

The annotator reads through the text or ASR output, and/or listens to the audio recording, and makes an initial YES/MAYBE/NO judgment about the suitability of a file for inclusion in the selection pool. The primary determination for suitability is based on topic content. Hard news about politics and current events, social issues, human interest stories, and opinions about current news are all examples of suitable topic content. This data may be collected as editorials, interviews, call-ins, round table discussions or from

simple news reports. There are many categories of topic content that is excluded. Some examples include: re-broadcasts or re-prints, previews, sporting events, lists of any kind, weather or stock reports, commercials, movie or book reviews, chain letters, fake news, and anything that contains explicit sexual or obscene language. Neither of these lists is intended to be exhaustive of all suitable and non-suitable topic content defined for use in GALE.

While collected source data varies in length, the GALE test set consists of smaller snippets - typically, segments under 200 words in length. (A snippet is defined as a single cohesive story or article contained in a much larger publication.) Because of this requirement, LDC annotators typically must designate one or more subsections of a given candidate file for selection. The toolkit includes functionality for marking off one or more segments (snippets) within an audio or text. When a segment is selected, the tool automatically updates the file inventory and file size (token or duration) information.

After the annotator labels a file or segment as YES or MAYBE, the tool launches a decision panel (see Figure 5.17) and the annotator records the following judgments:

- Genre: NW, WB, BN, BC, Other (with text box)
- Topic Category:
  1. Local or regional news - general
  2. National or international news - general
  3. Editorial or opinion
  4. Personal anecdote
  5. Human interest story
  6. Medical
  7. Legal
  8. Other
- Synopsis: 1-2 sentence description of content, in English
- Document difficulty rating[17]: Less than 2; 2; 3; 3+ (See description of ILR ratings in Section 5.3.3.)
- Language/Dialect issues: text box
- Other comments: text box

After making the relevant judgments, the annotator moves on to the next file (or segment of the same file) for review.

At the close of the work session, the annotator is asked by AWS whether the file is complete or still in progress. If the file is labeled complete, AWS will allow the annotator to request another file for review or to log out. If the file is labeled in progress, AWS will launch the same file in the selection tool the next time the annotator logs in for a work session.

[17] LDC annotators assign preliminary ratings using the Interagency Language Roundtable (ILR) system. Although there are no hard and fast requirements regarding document difficulty, the selection process primarily targets level 2 and level 3 documents. See http://www.dliflc.edu/academics/academic_affairs/dli_catalog/reading.htm

First Round Automatic Selection

Once manual selection on the candidate pool is complete, LDC annotators create quick rich transcripts for selected audio snippets, and segment the selected text snippets into sentence units, resulting in preliminary source text references (Part 1 in this book). LDC then distributes to NIST the preliminary source text references along with the output of the data selection database consisting of the following fields:

- Snippet ID
- Language
- Genre
- Source
- preliminary ILR
- Publication date
- Topic category
- Topic description
- Language / Dialect issues
- Document section location \{Begin, Middle, End\}
- Document section word count (text)
- Document section duration (audio)

The next stage of data selection applies a series of automatic diagnostics to calculate log-perplexity and 3-gram hit-rate for selected snippets.

First Round of Down Selection

Automatic measures of *n*-gram statistics and perplexity as measured against the original Phase 1 training data[18] were used as a first step in controlling for year-to-year test set differences. The evaluation candidate pool of potential test materials were sampled, removing candidate documents that were outside the Phase 1 test material statistics in either *n*-gram hit rates or log-perplexity. From the remaining set of documents, NIST semi-randomly selected a set that was three times the anticipated test set size such that the *n*-gram hit rate distribution and log-perplexity scores were similar to that of the phase 1 test sets.

The selected files (three times the intended test set) were returned to the LDC for further processing. Audio sources were processed with in-house speech-to-text systems to generate rough transcriptions. These transcripts and the source data from the text genres were processed by in-house MT systems to produce system translations. The MT output and the rough human translations were returned to NIST for the second round of down selection.

Second Round of Down Selection

[18] This down-selection process did not apply to phase 1 evaluation data.

Before Phase 2, the hope was that matching *n*-gram statistics and perplexity measures would be sufficient to produce stable, in terms of test set difficulty, succeeding year test sets[19]. It was not. In Phase 2, post evaluation analysis indicated that the Chinese test set was slightly more difficult than the Phase 1 test set which confounded the ability to demonstrate technology progress. Demonstrating progress is of vital importance in any evaluation that has strict targets of performance that must be met for continuation in the program. The fact that there was a measurable difference in test set difficulty was not a surprise and underscored the fact that inherent differences in the data exist despite best efforts to eliminate those differences. Language is ever changing.

A second round of down selection was implemented in Phase 3. In another attempt to reduce year-to-year test set differences the test material was carefully selected such that its overall TER distribution matched that of a previous year's test set. The idea is that an MT system is likely to produce similar TER distributions on two corpora with the same overall difficulty, provided that the corpora share the same genre, and that the system is not trained on either corpus. The use of system technology to select evaluation material is not a practice that NIST endorses. The same weaknesses that are a part of the system(s) being used to select the data will not be properly probed by the new test. Overriding the NIST concern is the need to demonstrate progress.

Along with the rough human translations of all the data being considered for test material, the LDC provided NIST with three system translations for each document, each produced by a different MT system. A per document TER score was established by averaging the scores from each MT system. In a similar manner TER scores were generated for each of the preceding evaluation test sets. The phase 3 test set was selected so that the TER distribution of the data matches the desired distribution. For the Arabic test materials, Phase 3 data was to match the TER distributions obtained in Phase 2. For Chinese test materials, Phase 3 data were to match the TER distributions obtained in phase 1.

This second round of down selection reduced the number of files to be submitted to Quality Control (QC) for final reference translation to the actual evaluation test set plus a handful of additional files, held in reserve, in case a file or two is deemed inappropriate for evaluation during the QC process.

While this second round of down selection was not performed to perfection in Phase 3, it is believed that in future phases of GALE this step will help reduce (but not eliminate) year-to-year test set differences. Short of using a progress test set - data that is kept blind and reused in future evaluations - language data is varied enough that efforts to control levels of difficulty will prove at times to be ineffective.

Final Reference QC

With the final selected files in hand, the LDC performs translation and QC, the last step in identifying possible issues with the selected data before declaring the official evaluation test set as finalized. The complete set of reference translations with

[19] This down-selection process did not apply to either of the first two phases of GALE. It has since become part of the data selection protocol and is planned for all future phases.

segmentation that matches the source files are provided to NIST for GALE MT evaluation, which involves HTER scoring of translations from the evaluation systems.

As described in Sections 5.2.3 and 5.4, HTER is a measure of edit distance between a modified system output and a single reference translation. Acknowledging that there are multiple acceptable translations HTER is designed to capture the quantity of edits required to obtain the correct "meaning." Sometimes the source data can be ambiguous as to the intended meaning. In such cases alternative translation are encoded in the reference translation allowing the post editor to use their best judgment as to which "meaning" the system is closest to capturing. The actual implementation of alternatives in translation is described above in Section 5.4.

The LDC has created documentation that defines the process used to create reference translations. These translation guidelines are available from the LDC web-site[20].

Control Documents

As stated in the preceding sections, despite best efforts, some level of difference will often exist between two test sets. The GALE program has made good use of *control documents* to assist in understanding test set differences. Control documents are used in two facets.

In the first situation, several documents from an evaluation are declared *sequestered* shortly after the evaluation period. Team interaction with sequestered data is limited, and no system development is permitted that makes use of information derived from the sequestered files. In the subsequent evaluation these sequestered documents will be reprocessed by the GALE teams along with the new evaluation test set. The data is post-edited and scored. While not a part of the official GLE evaluation, the scores obtained from the sequestered data alone are useful as a benchmark of improvement from the previous year. This does not represent a perfect solution since the remaining data from the previous evaluation are used to develop the new system and that data is drawn from the same epoch where similar language, topics, and names are used, which might create a bias.

In the second situation, control documents are used to measure year-to-year editor differences. See Section 5.4.3.3.5 for a description of how control documents are used to measure changes in editor behavior from one evaluation to the next.

5.4.4.5. Metrics

In addition to the official GALE MT evaluation metric of HTER, other automatic metrics that GALE teams might use in the development cycle are reported to the teams to assist in error analysis. The automated metrics reported by NIST include: BLEU (Section 5.2.2.2 [version 1.04] above), METEOR (Section 5.2.2.3 [version 0.6] above), and TER (Section 5.2.2.4 [version 0.7.25] above). NIST reports per-document HTER scores that are used by DARPA to determine the success of teams meeting DARPA GALE program's performance targets.

[20] http://projects.ldc.upenn.edu/gale/Translation/

5.4.4.6. Evaluation Rules

Each evaluation is implemented according to predefined rules and restrictions that identify the evaluation protocols. NIST documents these protocols in an evaluation specification document. The reader is referred to the NIST GALE web site for access to these evaluations specification documents[21] (evaluation plans). A few of the rules and protocols are listed here.

1. Teams are forbidden from using data created or published during the same time epoch that the evaluation data is chosen from.
2. Teams are forbidden from manually inspecting the evaluation data before their results are submitted to NIST for scoring.
3. Teams are required to share all self collected training data.
4. Teams are required to process the entire evaluation test set and return resulting translations to NIST by an agreed upon due date, usually allowing two weeks to process the text sources, and three weeks to process the audio sources.

5.4.4.7. Systems Evaluated

As identified earlier there are three GALE teams. Each team submits a single set of results for each of the two test sets (Arabic and Chinese). The team names are AGILE (led by BBN Technologies), ROSETTA (led by IBM), and NIGHTINGALE (led by SRI International).

5.4.4.8. MT Results

In this section we report the performance results obtained in each of the GALE phases. Section 5.6 focuses on the primary metric of HTER. Section 5.4.4.8.2 focuses on the automated metrics that were reported to the GALE teams. Section 5.4.4.8.3 focuses on the intermediate HTER scores calculated from the first pass and second pass post editing.

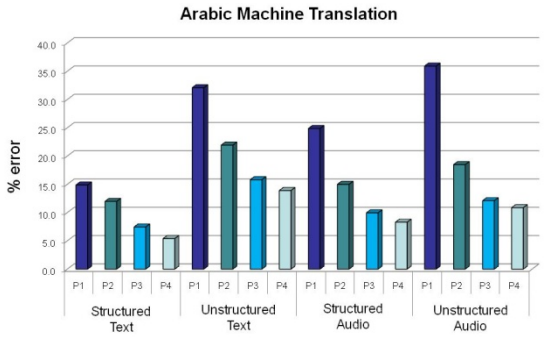


Figure 5.18: Average HTER for all genres of Arabic sources

[21] http://www.nist.gov/speech/tests/gale/

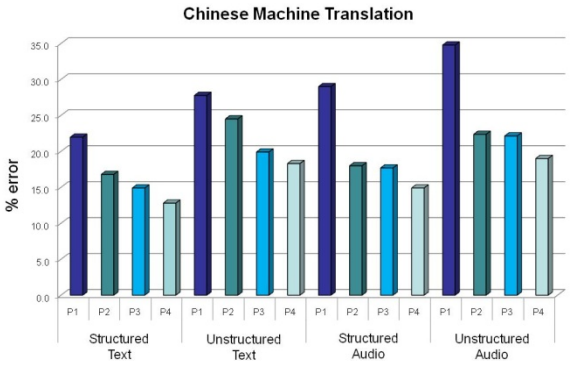


Figure 5.19: Average HTER for all genres of Chinese sources

5.4.4.8.1. HTER Results

The scores presented in this section represent the final HTER scores as calculated after completion of all post-editing. Scoring is divided into four genres: NW for newswire data, WB for web data, BN for broadcast news data, BC for broadcast conversation data. The results are shown in Figures 5.18 and 5.19.

These averages were only used for the first year of GALE. For the following years, DARPA changed the metric. Instead of aiming for an average translation accuracy rate, the program set goals based on a given percentage of documents that exceeded a minimum accuracy. The targets for years two to five are shown in Table 5.9

Language	Arabic				Chinese			
Genres	NW	WB	BN	BC	NW	WB	BN	BC
Phase 2	80/90	70/75	75/80	70/70	75/90	70/75	70/75	65/70
Phase 3	85/90	75/80	85/85	75/75	80/90	75/80	80/80	75/75
Phase 4	90/90	80/85	90/85	85/80	85/90	80/85	85/85	80/80
Phase 5	90/95	85/90	90/90	90/90	90/90	85/90	90/90	85/85

Table 5.9: GALE translation targets shown as accuracy/percent of documents

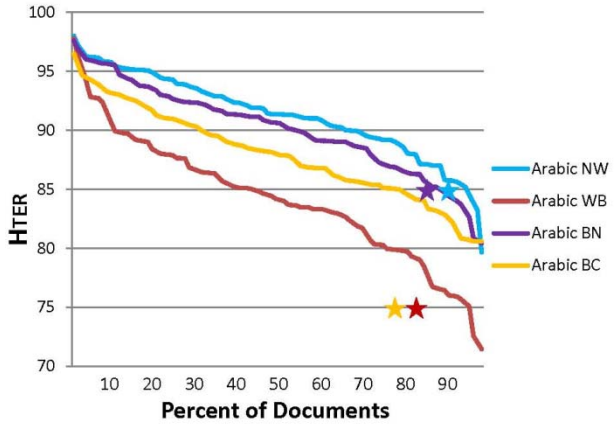


Figure 5.20: Arabic document accuracy plotted in descending order. Stars show the target for each Genre

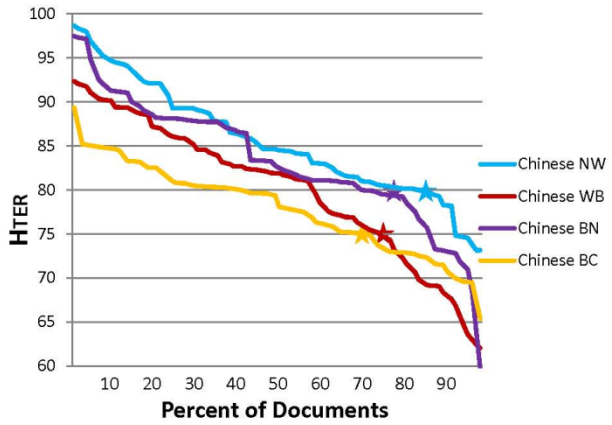


Figure 5.21: Chinese document accuracy plotted in descending order. Stars show the target for each Genre

Since the percentage of documents of a given accuracy was set as the GALE targets, the translation quality is best displayed in Figure 5.20 for Arabic and Figure 5.21 for Chinese. These figures display the document accuracies after they have been sorted. These are the best Phase 3 results. The figures show the Phase 3 targets as stars. All Arabic genre and three of four Chinese genres exceeded the targets. Only Chinese BN is slightly below target.

5.4.4.8.2. Automated Metric Results

Team	Genre	Average BLEU			Average METEOR			Average TER		
		P1	P2	P3	P1	P2	P3	P1	P2	P3
Agile	NW	0.23	0.33	0.4	0.55	0.62	0.67	0.55	0.45	0.4
	WB	0.11	0.22	0.27	0.40	0.53	0.57	0.69	0.58	0.52
	BN	0.16	0.27	0.35	0.48	0.58	0.64	0.64	0.53	0.43
	BC	0.14	0.23	0.28	0.44	0.55	0.57	0.68	0.57	0.49
Nightingale	NW	0.23	0.31	0.32	0.55	0.61	0.61	0.56	0.50	0.48
	WB	0.10	0.20	0.21	0.40	0.52	0.52	0.70	0.64	0.60
	BN	0.16	0.23	0.28	0.48	0.56	0.58	0.69	0.61	0.51
	BC	0.12	0.23		0.42	0.53		0.75	0.64	
Rosetta	NW	0.24	0.29	0.38	0.56	0.61	0.66	0.55	0.47	0.41
	WB	0.13	0.19	0.25	0.42	0.50	0.55	0.68	0.58	0.53
	BN	0.17	0.28	0.35	0.47	0.59	0.64	0.70	0.60	0.50
	BC	0.13	0.20	0.27	0.42	0.50	0.57	0.70	0.60	0.50

Table 5.10: Arabic: Summary of Automated Metric Scores across Phases for BLEU, METEOR, TER (error measure). For each phase the average metric score is shown.

Several automated metrics were run over the submitted systems data compared against the single reference. Since these metrics do not make good use of embedded alternatives, only the first listed in the set of alternatives are used. The translation

agencies are instructed to put the most common, or their best guess as to the intended meaning, first.

Team	Genre	Average BLEU			Average METEOR			Average TER		
		P1	P2	P3	P1	P2	P3	P1	P2	P3
Agile	NW	0.17	0.21	0.18	0.50	0.51	0.49	0.63	0.60	0.63
	WB	0.14	0.14	0.16	0.44	0.44	0.43	0.65	0.66	0.66
	BN	0.13	0.21	0.15	0.45	0.51	0.46	0.75	0.59	0.65
	BC	0.10	0.11	0.11	0.39	0.39	0.39	0.77	0.71	0.69
Nightingale	NW	0.14	0.17	0.16	0.48	0.49	0.47	0.67	0.64	0.65
	WB	0.11	0.13	0.13	0.41	0.42	0.42	0.69	0.68	0.67
	BN	0.12	0.19	0.14	0.43	0.49	0.45	0.73	0.62	0.66
	BC	0.08	0.11	0.11	0.37	0.37	0.39	0.78	0.70	0.68
Rosetta	NW	0.14	0.17		0.45	0.48		0.67	0.64	
	WB	0.10	0.12		0.39	0.40		0.71	0.68	
	BN	0.08	0.16		0.38	0.46		0.76	0.62	
	BC	0.06	0.10		0.33	0.36		0.79	0.70	

Table 5.11: Chinese: Summary of Automated Metric Scores across Phases for BLEU, METEOR, TER (error measure). For each phase the average metric score is shown.

Tables 5.10 and 5.11 summarize the average scores for three popular automated metrics, for Arabic and Chinese, respectively. Please note that TER is an error measure so its polarity is different from the other two.

5.4.4.8.3. Correlation Study

Correlation results are generally in MT evaluation to compare automatic and human evaluation of output from various machine translation systems. In this section, we present a correlation analysis of the results obtained from comparing each of the three engine outputs for the GALE data across evaluations, source language, and genre. We choose to use Spearman correlation as our measure for the strength of relationships between each of the variables of interest. Spearman rank correlation is a distribution-free rank statistic that tests the direction and strength of the relationship between two variables (Lehmann and D'Abrera 1998). This method, based on ranking the two variables, makes fewer assumptions about the distribution of the values and measures monotone rather than linear covariation. We show the relationship between automated metrics BLEU, NIST, TER, and METEOR) and both the human judgments of quality (as denoted by adequacy scores) and HTER (the official metric of GALE). Both automated metric scores and human judgments are calculated per segment and then correlated at that level. Adequacy scores amongst raters are averaged prior to performing correlations.

A. Analysis and Results

First, we computed correlation scores for the entire data set. Table 5.12 shows these results. The results presented are the absolute value correlations between the variables. This is done because as TER is an edit distance measure (smaller score means better), the true correlation between it and similarity metrics (larger score means better) has a reverse relationship. METEOR and HTER are found to correlate highest with adequacy as compared to the other metrics. As should be expected, TER correlate highest with HTER.

	HTER	BLEU	NIST	TER	METEOR
a)	0.622	0.508	0.577	0.517	0.624
b)	----	0.502	0.473	0.528	0.512

Table 5.12: Automated Metric Correlation with (a) Adequacy judgments and (b) HTER scores

	HTER	HTER	BLEU	NIST	TER	METEOR
a)	Arabic	0.639	0.518	0.558	0.520	0.611
	Chinese	0.584	0.431	0.520	0.438	0.570
b)	Arabic	----	0.592	0.588	0.626	0.614
	Chinese	----	0.399	0.364	0.428	0.417

Table 5.13: Automated Metric Correlation with (a) Adequacy judgments and (b) HTER scores classified by Source Language.

We see in Table 5.13 that there is a change in the relationship between automated metrics and adequacy by source language. Within each language, HTER has higher correlation with adequacy scores. Across the board, scores are lower for Chinese than for Arabic. Similar findings have been confirmed in previous NIST MT evaluations.

Similar analysis shows a dramatic decrease in the magnitude of correlation within the unstructured data track of broadcast conversation across most metrics. With the exception of the broadcast news track, METEOR presents a stronger relationship with human judgment within genres.

B. Discussion of the Correlation Process

Correlation helps establish whether there is a relationship between machine translation output quality as determined by commonly used machine translation evaluation metrics and human induced adequacy or HTER judgments. What correlation does not provide is a correct sense of predictability of one for the other, i.e. it does not answer the question: Can automated metrics be used instead of human judgments? To correctly characterize a predictive relationship, we would need to extend beyond correlation and determine which automated metrics would best serve as predictors in distinguishing adequacy or HTER. Statistical modeling techniques would be useful in this direction.

C. Section Summary and Conclusion

We have shown various correlation results for several cross-classifications of the GALE data. In most instances, HTER and METEOR tend to be closer related to human judgments of adequacy the other automated evaluation metrics. Because HTER scores are just as time consuming to reproduce as adequacy metrics, it would be beneficial to find a metric that not only correlates well with adequacy but also HTER. While TER generally correlates better with HTER, it does not outperform other metrics in comparison to adequacy scores. Hence, METEOR seems to be the best automated metric in terms of assessing both adequacy and HTER. Though it does not correlate well to HTER, BLEU scoring remains the standard method for most internal development because of its low cost and rapid turnaround time.

Chapter 5.5 Other Use for Evaluations

5.5.1. Effect of MT Evaluation Measures on Optimization

Authors: Arne Mauser, Saša Hasan, Alok Parlikar, Stephan Vogel

5.5.1.1. Introduction

Och (2003) suggested a procedure to automate research in MT with the help of a numerical optimization technique that works with an error surface on the MT output as defined by an automatic evaluation metric such as BLEU or TER. The error surface as defined in the space of model weights is known to be very bumpy. It has several sharp hills and valleys, and many plateaus. Numerical optimization can only guarantee a local optimum in this space. The final optimal value depends on where we start in this space, and also how the optimization algorithm searches the space. It has been shown that using random restarts in the weight space during search typically gives better optima (Moore and Quirk 2008). However, the search still remains a complex problem, which leads to a number of questions.

- How does selection of the tuning set impact the performance on unseen test data?
- How does optimization on one evaluation metric affect the performance on other metrics?
- Do different numerical optimization algorithms lead to different optima? How robust are these methods to the initial search point?
- Should all training samples be given the same importance or should, for example, optimization be biased to improve bad translations more?

Also note that any optimum found is still determined by an automatic metric, which may or may not reflect how humans actually perceive the optimized output.

In the following sections, we will explore these questions in a number of different experiments. Before doing so, we will first take a quick look at how MT systems are typically composed of a log-linear combination of models. We will also review the process of automatic optimization, and look at different numerical optimization techniques.

5.5.1.2. Minimum Error Rate Training

Minimum error rate training (MERT) is essentially a method for automatically optimizing the weights for each of our models (see Part 2) to achieve optimal performance. This is typically done in three steps: (i) choose an automatic evaluation metric that can judge how good or bad a particular translation output is; (ii) define an error function on the entire MT output with the help of this metric; and (iii) use a numerical optimization method to minimize this error function.

MERT can deal with several types of output formats from an MT system. The most common strategy is to use an n-best list of hypotheses. Given a vector of model weights, we can re-rank the hypotheses for each sentence in the n-best list and establish a new

first-best translation for those weights. Thus, the error function is defined by the metric error on the new first-best hypothesis for a given weight vector. Some MT systems can output an entire lattice of translations instead of just the n-best hypotheses. Lattices are compact representations of several hypotheses, so they contain more alternatives than what we would have in an n-best list. However, an error function on a lattice could be defined in a similar manner: for each sentence, choose the best hypothesis from the lattice, and the metric then defines the corresponding error.

The error function thus defined has several properties. First, it is not a convex function; it has many local optima. Secondly, the function is not necessarily differentiable or even continuous in the weight space, and a gradient function on the error surface cannot be analytically defined. Two widely used optimization techniques, which are applicable under such conditions, are the Downhill Simplex method and the Powell method.

5.5.1.3. Downhill Simplex

The Downhill Simplex method (Nelder 1965) is a commonly used nonlinear optimization algorithm. Given an N-dimensional space to search in, the method uses an N+1 dimensional polyhedron, called the simplex. The error is evaluated on each vertex of the simplex. The vertex with the highest error is discarded and a new test position is generated based on the behavior of the error on the different vertices. To avoid getting stuck in a local optimum and to achieve convergence, the downhill simplex algorithm includes a shrinking of the simplex.

5.5.1.4. Powell

The Powell method (Powell 1964) is another commonly used optimization method. This method uses a set of directions in the search space. Starting from an initial guess, the method moves along one dimension until a minimum is reached, then continues moving along the next direction until it finds the minimum, and thus cycles through the set of directions until the error function in minimized. It then modifies the set of directions, and repeats the process, until convergence is achieved.

5.5.1.5. Experiments

We conducted our experiments on the NIST MT Chinese-to-English task. Details are provided below regarding data characteristics, generalization to other data sets and metrics, and comparison of optimization techniques.

Data

The international evaluation held by the US National Institute of Standards and Technology (NIST) is focused on news translation of Arabic and Chinese to English. The bilingual training is provided by the Linguistic Data Consortium (LDC) and consists of newswire and news magazine translations, UN documents and parliamentary proceedings. In total, we have about 8 million sentence pairs or 250 million running words.

The evaluation corpus from 2002 is used as main development set. In most experiments we optimize the system weights on this corpus. The years 2003 to 2005 serve as test sets unless stated otherwise. Each corpus consists of about 600 to 1000 sentences and has 4 reference translations. All measures are computed disregarding the case of test. Evaluation of the corpus from the years 2003 to 2005 shows a degradation of 1.8 to 3 points in the BLEU score and a corresponding degradation in all other metrics evaluated (TER, WER, PER and NIST)

Generalization to other Metrics

Different MT evaluation metrics favor different features in the generated translations. For example TER typically favors shorter translations than BLEU. Therefore, optimization using one metric will not give optimal performance when evaluating using other metrics (Och 2003). However, if our MT system improve using one or two metrics, but degrades evaluating with other metrics, we can hardly feel confident that the translation quality really improved. Preferably, we would like to see an improvement on a number of different MT evaluation metrics.

	Evaluation					
Opt. on ↓	BLEU	TER	WER	PER	NIST	Avg. Len.
BLEU	35.9	56.7	63.1	39.7	9.63	31.8
TER	34.6	55.7	62.0	40.4	9.41	29.2
WER	33.2	55.5	61.0	41.8	8.93	27.4
PER	35.1	57.3	64.5	39.8	9.59	31.9
NIST	35.8	56.2	63.1	39.5	9.66	31.3
BLEU+ TER	35.4	55.8	62.2	39.8	9.56	30.2

Table 5.14: Error rates on the NIST 2005 corpus when optimizing an evaluating with different measures. TER, WER, PER: lower values are better, BLEU, NIST: higher values are better

In order to find a good overall criterion for system tuning, we examined the effect of optimizing on one measure and evaluating on all. The results for the NIST 2005 set are shown in Table 5.14. This table shows that the best results are obtained when evaluating with the same or similar metrics to the one used for training (BLEU, PER, and NIST vs. TER and WER), using non similar metrics shows some degradation. This grouping also shows a large difference in the sentence length. The word count for BLEU, NIST and PER shows longer hypotheses. TER and especially WER lead to rather short hypotheses.

Since the sentence length is rather different, we also tried to optimize on the sum of error-rate version of BLEU (100- BLEU) and TER. The result shows a good performance on most error measures, indicating that it could serve as a reasonable all-round criterion.

Comparing Simplex and Powell

Different optimization techniques will typically find different optima, even when starting from the same initial configuration. And starting from different initial configurations can lead to very different local optima. It is, therefore, advisable to use multiple random restarts (Moore and Quirk 2008). However, while finding a good local optimum is the primary goal, it is also important to find it fast. Some evaluation metrics are computationally expensive. They can only be used in automatic tuning if the

optimization procedure converges within a small number of steps, i.e., if only a small number of different hypotheses need to be evaluated.

We therefore studied how the two numerical optimization algorithms, Simplex and Powell, compare to each other in terms of speed of convergence and quality of the results. For this comparison we used the implementation of these algorithms from the SciPy toolkit. In this experiment, we used a 1500-best list of hypotheses on the MT05 test set for Chinese-English translation task. We separately used the TER and BLEU metrics to define the error function for optimization.

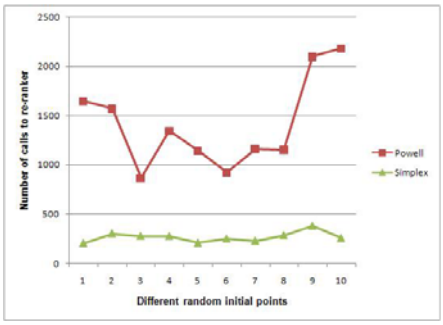


Figure 5.22: Comparison Powell vs. Simplex: Number of calls to re-ranker for different random starts.

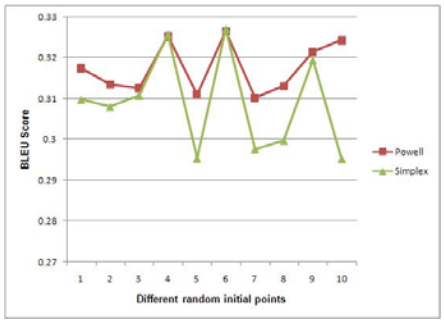


Figure 5.23: Comparison Powell vs. Simplex: Bleu scores for different random starts.

We chose ten different random points in the weight space as starting configurations and we ran both the Simplex and Powell methods to find the optimal value. We looked at what the resulting error value was, and how many different points in the weight space were evaluated before convergence. Ideally, we would like to have as low an error as possible, with as few as possible calls to the error evaluation. Note that error evaluation requires re-ranking an entire n-best list, and depending on the number of sentences in our test set and the size of the list, this operation can add significantly to the overall processing time. The comparison of the number of re-ranker calls required is shown in Figure 5.22. The corresponding BLEU scores achieved at the end of optimization starting from those random points is shown in Figure 5.23.

Although these figures are based on error function defined by the BLEU metric, a similar observation is made upon using TER. We observe that Powell's method consistently made many more function evaluations than simplex, and usually also ended up with a lower error than simplex method. We should note, however, that the difference between the Powell and Simplex methods, in terms of best BLEU scores, is rather small (0.008 average), and Simplex is much faster. I.e., we can run Simplex with more random restarts.

5.5.1.6. Document Level Tuning

MT systems are generally optimized towards the corpus level score of a dev-set. We have run some initial experiments to see what happens when we tune towards a collection do documents, each of which are weighted differently. The idea here is to investigate

whether we can improve on hard-to-translate documents, by giving them more weight during optimization.

In our setup, we specify how many classes of documents we would like to have and we specify a weight for each class. After one round of decoding, we rank documents based on the BLEU scores of their top-best translations. We then chunk the ranked list of documents into stipulated classes. Based on what weight a document has, the metric features (*n*-gram counts in case of BLEU) for every sentence in the document are scaled up or down in the n-best list. MERT is run on this scaled n-best list, to find model weights that are then used in the next decoding iteration. Note that documents are re-ranked and reclassified in every MERT iteration, so they can have different weights every time.

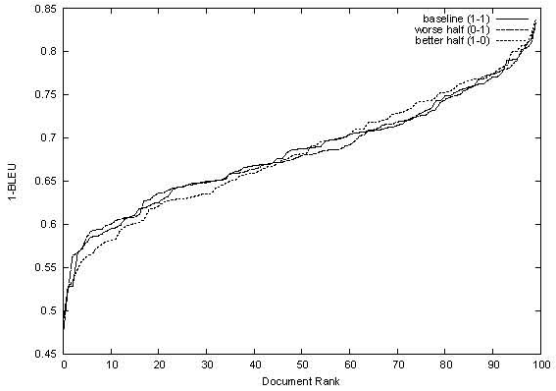


Figure 5.24: Document Scores on Dev Set

We ran initial experiments to understand how this document level tuning behaves. In one setup, we set the class weights to '1,0', which means only the top 50% best translated documents participate in the optimization. In the other experiment, we set the weights to '0,1', so only the worse half of the corpus participates in the optimization. Our baseline system uses all the documents, which is equivalent to having weights '1,1'. The dev-set used here is the NIST-MT05 set for Chinese-English translation. After four iterations of decode-MERT, we plotted the ranked errors of documents in each of the setups. We see that the error curve tilts in favor of the half that was used in tuning. Figure 5.24 shows this trend.

The scores of these three systems on an unseen test set, part of the DEV-07 set containing 588 sentences with one reference each. Show that tuning towards bad documents did not change the scores, but tuning to just the better half of the documents had a deleterious impact on both average scores, and the tail.

From these preliminary experiments, our hypothesis is that including poorly performing documents from the tuning set in optimization is important. We believe that assigning appropriate weights to the documents based on their scores can lead us to overall improvements, and we are investigating this. We notice that between the different MERT iterations, documents jump classes. Having a rigid 0-1 boundary is not desirable, because having a largely different set of documents in MERT could make it brittle.

Instead, we plan to use a smoother weight distribution - using more than just two classes, and assigning weights in steps.

5.5.1.7. Conclusions

In this section, we looked at the automatic optimization of MT systems. We went over the Powell and Simplex methods of numerical optimization. Our experiments showed that Powell is more robust to random variation in the starting point in our setup. We presented a comparison of how tuning towards particular evaluation metric affects the performance along other metrics. We observed that BLEU and TER are robust metrics for optimization, because of two reasons. First, improvement on a development set results in improvement on similar test sets, and second, the improvement is seen consistently along all metrics of evaluation. This is not the case for metrics such as WER and PER, and hence using either BLEU, or TER, or an interpolation of these two metrics seems to be an overall good choice for tuning an MT system.

Chapter 5.6 Searching for Better Automatic MT Metrics

Authors: Alon Lavie, Abhaya Agarwal, Michael Denkowski, Matthew Snover, Nitin Madnani, Bonnie Dorr, Richard Schwartz, Nizar Habash, Jeremy Kahn, Mari Ostendorf, Brian Roark, Seth Kulick, Mitch Marcus, Sebastian Pado, Michel Galley and Christopher Manning

5.6.1. Introduction

This section describes recent research work by members of the GALE community on developing advanced automatic evaluation metrics for machine translation. The primary goal behind most of these efforts is to develop metrics with high levels of correlation between automatic metric scores and human judgments of translation quality. The commonly used metrics to date, previously described in section 5.2.2, while effective for ranking systems, do not consistently correlate highly with human judgments of translation quality at the granularity levels of individual segments and documents. These levels are of particular interest for developers of MT systems within GALE, since the target goals of the GALE evaluations are formulated around percentiles of documents that score higher than set thresholds, as calculated by HTER scores. The new metrics described in this section are designed to address this fundamental issue. In some cases, the metric developers explicitly aim to directly improve correlation levels with HTER. This would provide MT system developers with fully automatic metrics for system evaluation and parameter optimization that are more effective in maximizing system performance, as measured by HTER.

We review six metrics in this section. While the primary goals of these metrics are fundamentally similar, the means by which they seek to achieve these goals are quite different. There are, however, several consistent themes. Several of the metrics (METEOR, TERP, RTE) seek improvements by addressing the language variability issue, where the same meaning can be expressed in various ways using different words and constructs. To address this variability, these metrics employ methods for detecting the

similarity between an MT-produced translation and a reference at the word level (synonyms) and/or the phrasal level (paraphrases). A second common theme to several of the metrics (SEPIA, EDPMEmpty, MULCH, RTE) is their focus on syntactic features, in addition to, or instead of, features at the word and *n*-gram level. The rationale behind this is that similarity in meaning between an MT-produced translation and a reference may be more accurately identified by syntactic components of the two sentences, rather than correspondences at the shallow lexical-level. A third strong emerging theme for several of the metrics (METEOR, TERP, RTE) is their use of Machine Learning techniques for parameter optimization in order to obtain maximum correlation with human judgments. These metrics combine several features and components, which can be assigned relative weights. The idea behind this optimization process is similar to the goal of using automatic MT metrics for optimizing the performance of MT systems. In this case, however, the parameters of the metric itself are optimized to correlate best with human judgments on sets of data. Interestingly, this results in a two-stage optimization chain, where an automatic metric is first optimized against human judgment data, and the resulting metric is then used to optimize MT systems. This gets around the difficult problem of optimizing MT systems directly against human judgments, which are difficult and expensive to obtain in quantities that would be required for direct optimization of the MT systems.

In the sub-sections below, we describe six different recently proposed automatic MT evaluation metrics. The descriptions include some initial assessments of improvements provided by the metrics in comparison with previous state-of-the-art metrics. All but one (MULCH) of these six metrics participated in the NIST MetricsMATR evaluation - an open comparative evaluation of automatic metrics for MT, which was organized by NIST in late 2008. Detailed contrastive results analyzing the performance of these and other metrics can be found on the NIST website[22].

5.6.2. METEOR

The METEOR metric (Lavie 2004; Banerjee 2005) was originally developed in 2004, and the commonly used version of METEOR was already briefly described in Section 5.2.2.3. Here we describe some recent work on improving the metric and its correlation with human judgments. Much of the recent work on METEOR has focused on the tuning and optimization of the three adjustable parameters within the metric. This involved experimenting with optimizing the parameters on different data sets and to several different formulations of human judgments. The subsection below recaps the parameter optimization process used and presents the results of these optimization experiments.

5.6.2.1. Optimizing Metric Parameters

The original version of METEOR (Banerjee 2005) defines instantiated values for three parameters in the metric: one for controlling the relative weight of precision and recall in computing the Fmean score (α); one governing the shape of the penalty as a function of fragmentation (β) and one for the relative weight assigned to the fragmentation penalty

[22] http://www.itl.nist.gov/iad/mig//tests/metricsmatr/

(γ). For details, see the METEOR description in Section 5.2.2.3. In the original version of METEOR, these parameters were instantiated with the values α=0.9, β=3.0 and γ=0.5, based on early data experimentation.

Over the past two years, in the context of the contrastive evaluations of automatic MT evaluation metrics conducted by the WMT-07, WMT-08, and the NIST MetricsMATR workshops, a thorough investigation was conducted aimed at tuning these parameters based on several available data sets, with the goal of finding parameter settings that maximize correlation with human judgments. Human judgments traditionally come in the form of ``adequacy'' and ``fluency'' quantitative scores. More recently, a new form of human judgments in the form of binary and n-ary human rankings of translation output have also become available. The experiments conducted looked at optimizing parameters for each of these human judgment types separately, as well as optimizing parameters for the sum of adequacy and fluency. Parameter adaptation is also an issue in instances of METEOR that have been created for other languages, since parameters that were optimized to maximize correlation with human judgments for English would not necessarily be optimal for other languages.

In all of these optimization experiments, ``hill climbing'' search is used to find the parameters that achieve maximum correlation with human judgments on the training set. Pearson's correlation coefficient was used as the measure of correlation. A ``leave one out'' training procedure was used in order to avoid over-fitting. Whenever training data segments from *n* MT systems was available for a particular language, parameters were trained *n* times, leaving the data from one system out in each training, and pooling the segments from all other systems. The final parameter values were calculated as the mean of the *n* sets of trained parameters that were obtained. When evaluating a set of parameters on test data, segment-level correlations with human judgments are computed for each of the systems in the test set and the mean over all systems is then reported.

5.6.2.2. Optimizing for Adequacy and Fluency

Parameters were trained to obtain maximum correlation with normalized adequacy and fluency judgments separately and also trained for maximal correlation with the sum of the two. To partially address human bias issues, the human judgments were *normalized*, which transforms the raw judgment scores so that they have similar distributions. The normalization method used is described in (Blatz 2003). Multiple judgments were combined into a single number by taking their average. A detailed description of these experiments appears in (Lavie 2007}.

	Adequacy	Fluency	Sum
α	0.82	0.78	0.81
β	1.0	0.75	0.83
γ	0.21	0.38	0.28

Table 5.15: Optimal Values of Tuned Parameters for Different Criteria for English

The resulting optimal parameter values on the training corpus are shown in Table 5.15. The optimal parameter values found are somewhat different than the previous settings of the metric parameters (lower values for all three parameters). The new

parameters result in small, but noticeable improvements in correlation with human judgments on both training and testing data. Tests for statistical significance using bootstrap sampling indicate that the differences in correlation levels are all significant at the 95% level. An interesting observation is that precision receives slightly more ``weight'' when optimizing correlation with fluency judgments (versus when optimizing correlation with adequacy). Recall, however, is still given more weight than precision. Another interesting observation is that the value of γ is higher for fluency optimization. Since the fragmentation penalty reflects word-ordering, which is closely related to fluency, these results are consistent with expectations. When optimizing correlation with the sum of adequacy and fluency, optimal values fall in between the values found for adequacy and fluency.

5.6.2.3. Optimizing for Rankings

Callison-Burch (2007) reported that the intercoder agreement on the task of assigning ranks to a given set of candidate hypotheses is much better than the intercoder agreement on the task of assigning a score to a hypothesis in isolation. Based on that finding, in WMT-08, only ranking judgments were collected from the human judges.

A new set of experiments was therefore conducted to re-optimize the METEOR parameters in order to maximize for correlation with ranking judgments. This required computing full rankings according to the metric and the humans and then computing a suitable correlation measure on those rankings. METEOR assigns a score between 0 and 1 to every translation hypothesis. This score can be converted to rankings trivially by assuming that a higher score indicates a better hypothesis. In development data, human rankings were available as binary judgments indicating the preferred hypothesis between a given pair. There were also cases where both the hypotheses in the pair are judged to be equal.

In experiments conducted for the WMT-08 evaluation, the binary ranking judgments were converted into full rankings of all translations of each source sentence. Spearman correlation was then calculated between these human rankings and METEOR scores rankings for each source sentence. The final correlation score is the average of the Spearman correlations for all individual sentences. An exhaustive grid search was performed in the feasible ranges of parameter values, looking for parameters that maximize the average Spearman correlation over the training data. To get a fair estimate of performance, three fold cross validation was performed on the development data. Final parameter values were chosen as the best performing set on the data pooled from all the folds. (see Agarwal 2008).

In further experiments for the NIST MetricsMATR evaluation, a slightly different optimization process was used. Binary human rankings were not converted into full rankings. Instead, the metric parameters were directly optimized for maximizing the number of correct binary human rankings across the entire training data set.

	A	β	γ	Original	Re-tuned
English	0.95	0.5	0.45	0.3813	0.4020
German	0.90	3.00	0.15	0.2166	0.2838
French	0.90	0.5	0.55	0.2992	0.3640

Spanish	0.90	0.5	0.55	0.2021	0.2186

Table 5.16: Optimal Values of Ranking-Tuned Parameters for Various Languages and Average Spearman Correlation with Human Rankings for METEOR on Development Data

The re-tuned parameter values from the WMT-08 experiments and the average Spearman correlations for various languages with original and re-tuned parameters are shown in Table 5.16. The parameters show that for optimal correlation with ranking judgments the "Precision/Recall" balance is weighed almost completely toward "Recall", and this is consistent for all languages that we tested. Also, the re-tuned parameters for all the languages except German are quite similar. Comparing original and retuned parameters, we see significant improvements for all the languages. Gains are specially pronounced for German and French.

Overall, Meteor was one of the best performing metrics in the NIST MetricsMATR evaluation. Three versions of Meteor, differing only in their parameter settings, were evaluated. Meteor was one of the three top performing metrics in just about all conditions evaluating correlation with human judgments at the segment and document levels, for both adequacy and ranking forms of human judgments. Further details can be found on the NIST MetricsMATR website.

5.6.3. TER-Plus

While Translation Edit Rate (TER) (Snover 2004) has been shown to correlate well with human judgments of translation quality, it has several flaws, including the use of only a single reference translation and the measuring of similarity only by exact word matches between the hypothesis and the reference. These flaws are addressed through the use of Human-Mediated TER (HTER), but are not captured by the automatic metric. The handicap of using a single reference can be addressed by the construction of a lattice of reference translations, and such a technique has been used to combine the output of multiple translation systems (Rosti 2007). TERP does not utilize this methodology and instead focuses on addressing the exact matching flaw of TER.

TER-Plus (TERP)[23] is an extension of TER that aligns words in the hypothesis and reference not only if they are exact matches but also if they are determined to be related morphologically or to be synonyms, as well as directly aligning multi-word phrases by considering possible paraphrases of the reference words. Matching using stems and synonyms (Banerjee 2005) and using paraphrases (Zhou 2006; Kauchak 2006) have been shown to be beneficial for automatic MT evaluation. Paraphrases have also been shown to be useful in expanding the number of references used for parameter tuning (Madnani 2007; Madnani 2008) although they are not used directly in this fashion within TERP.

TERP uses all the edit operations of TER – Matches, Insertions, Deletions, Substitutions and Shifts, as well as, three new edit operations: Stem Matches, Synonym Matches and Phrase Substitutions. TERP identifies words in the hypothesis and reference that share the same stem using the Porter stemming algorithm (Porter 1980). Two words are determined to be synonyms if they share the same synonym set according to WordNet

[23] The name Terp doubles as the nickname--"Terp"—of the University of Maryland, College Park, mascot: the Diamondback Terrapin.

(Fellbaum 1998). Sequences of words in the reference are considered to be paraphrases of a sequence of words in the hypothesis if that phrase pair occurs in the TERP phrase table. Unlike TER, where all edits are given equal cost, the edit costs in TERP are optimized to maximize correlation with human judgments, and can be tuned to any particular type of human judgment.

5.6.3.1. Paraphrase Generation

TERP uses probabilistic phrasal substitutions to align phrases in the hypothesis with phrases in the reference. It does so by looking up – in a pre-computed phrase table – paraphrases of phrases in the reference and using its associated edit cost as the cost of performing a match against the hypothesis. The paraphrases used in TERP are extracted using the pivot-based method (Bannard 2005) with several additional filtering mechanisms to increase the precision. The corpus used for extraction was an Arabic-English newswire bitext containing a million sentences. Here are a few examples paraphrases that were actually used in a run of TERP:

controversy over → polemic about
by using power → by force
response → reaction

A probability for each paraphrase pair is computed as described by (Bannard 2005). The phrase table for TERP contains 14,184,361 paraphrases. Paraphrases are only used if the reference side of the paraphrase occurs exactly in the reference translation, allowing us to filter the paraphrase phrase table according to a reference set if needed.

5.6.3.2. TERP Edit Cost Optimization

In total, TERP uses 11 parameters (Match, Insert, Deletion, Substitution, Stem, Synonym, Shift and Phase Substitution) out of which 4 represent the cost of phrasal substitutions. These costs are optimized to maximize correlation with human judgments. The match cost is held fixed at 0, so that only the 10 other parameters can vary during optimization. All edit costs, except for the phrasal substitution parameters, are also restricted to be positive. A simple hill-climbing search is used to optimize the edit costs by maximizing the segment-level correlation of human judgments with the TERP score.

With the exception of the phrase substitutions, the edit cost for all other edit operations is the same regardless of what the words in question are. That is, once the edit cost of an operation is determined via optimization, that operation costs the same no matter what words are under consideration. The cost of a phrase substitution, on the other hand, is a function of the probability of the paraphrase and the number of edits needed to align the two phrases according to TERP. Specifically, the cost of a phrase substitution between the reference phrase, p_1 and the hypothesis phrase p_2 is:

$$\begin{aligned}\text{cost}(p_1, p_2) = \; & w_1 + \\ & w_2 \,\text{edit}(p_1, p_2)\; \log(\Pr(p_1, p_2)) + \\ & w_3 \text{edit}(p_1, p_2) \Pr(p_1, p_2) +\end{aligned}$$

$$w_4 \text{edit}(p_1, p_2)$$

(5.9)

editwhere $\text{edit}(p_1, p_2)$ is the number of edits according to TERP of aligning p_1 to p_2 and $\Pr(p_1, p_2)$ is the probability of paraphrasing p_1 as p_2, obtained from the TERP phrase table. Only paraphrases specified in the TERP phrase table are considered for phrase substitutions. In addition, the cost for a phrasal substitution is limited to values greater than or equal to 0, i.e., the substitution cost cannot be negative. The shifting constraints of TERP are also relaxed to allow shifting of paraphrases, stems, and synonyms.

5.6.3.3. TERp Alignment

In addition to providing a score indicating the quality of a translation, TERP also generates an alignment between the hypothesis and the reference, indicating which words are correct, incorrect, misplaced, or are close to the reference translation.

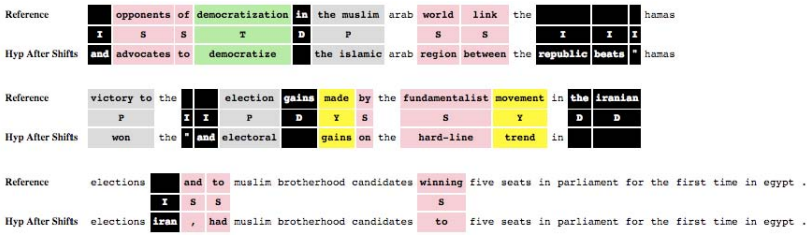


Figure 5.25: Example of TERP HTML Alignment Output

Consider an example MT output from the NIST MetricsMATR MT-06 data set and the closest of the four reference translations. A portion of the HTML output of the alignment generated by TERP is shown in Figure 5.25. The alignment shown is the final alignment after all shifts are performed. Each word or phrase in the hypothesis is aligned to a word or phrase in the reference, with the symbol between the word or phrases indicating the type of edit: "I" for insertions, "D" for deletions, "S" for substitutions, "T" for stem matches, "Y" for synonym matches, and "P" for phrasal substitutions. The lack of a symbol indicates an exact match.

5.6.3.4. Correlation Results

	Optimization set			Test set			Optimization + Test		
	Seg	Doc	Sys	Seg	Doc	Sys	Seg	Doc	Sys
BLEU	0.623	0.867	0.952	0.563	0.852	**0.948**	0.603	0.861	0.954
METEOR	0.731	**0.894**	0.952	0.751	**0.904**	**0.957**	0.739	**0.898**	0.958
TER	-0.609	-0.864	-0.957	-0.607	-0.860	**-0.959**	-0.609	-0.863	-0.961
TERP	**-0.782**	**-0.912**	**-0.996**	**-0.787**	**-0.918**	**-0.985**	**-0.784**	**-0.914**	**-0.994**

Table 5.17: MATR MT06 Pearson Correlation Results.

A portion of the MT-06 data was annotated with adequacy judgments for the 2008 NIST MetricsMATR evaluation (Przybocki 2008) and distributed to participants as

development data. TERP was optimized to maximize segment level Pearson correlation with Adequacy on two-thirds of this data (the Optimization Set). In addition, we also show the actual Pearson correlation numbers on this optimization set and the remaining portion of the development data (the Test Set) in Table 5.17. For comparison, correlations from the standard TER, BLEU and METEOR are also shown. Those correlations that statistically indistinguishable from the top metric (according to a 95% confidence interval) in each column are shown in **bold**. TERP had the highest correlation in each test case, and more importantly, showed significant improvement over the base TER evaluation metric.

5.6.4. SEPIA: Using Surface Span in Syntax-aware MT Evaluation

SEPIA is a newly developed MT evaluation metric that falls within the class of syntactically-aware evaluation metrics (Liu and Gildea 2005; Owczarzak 2007; Gimenez-Marquez 2007). The basic assumption of these metrics is that higher agreement in syntactic structure between MT output and Human reference translations reflects a higher degree of MT grammaticality and thus provides better correlation with human judgment than simple surface *n*-gram measures. An important failing in this assumption is that surface *n*-grams *do* capture a lot of syntactic information, especially for local dependencies. For instance, within the framework of dependency syntactic representation, 77% of all structural bigrams (parent-child links) have a surface span of less than 4 words and an absolute 47% of these are surface bigrams. Given this observation, SEPIA was designed with the goal of assigning bigger weight to structural bigrams with longer surface spans. Specifically, SEPIA uses a dependency representation but extends it to include surface span as a factor in the evaluation score. The dependency surface span is the surface distance between two words that are in a direct relationship in a dependency tree.

The rest of this sub-section describes the SEPIA metric and its variants, and summarizes how SEPIA performed in the recent NIST MetricsMATR evaluation (Habash 2008 SEPIA).

5.6.4.1. SEPIA

SEPIA evaluates a translation hypothesis segment (sentence) by computing a score based on a brevity-penalty-adjusted mean of multiple modified precision-based sub-scores. SEPIA uses two types of sub-scores: surface *n*-gram precision sub-scores (similar to BLEU (Papineni 2002)) and span-extended structural bigram precision sub-scores. We next discuss the latter type of sub-scores which are unique to SEPIA.

Span-Extended Structural Bigram Precision. A structural bigram (*SB*) is defined as a head word chain of size 2 (heads) in a dependency representation of the hypothesis/reference sentence. For example, in Figure 5.26, the edges linking the words *Among-crises*, *mentioned-Among* and *mentioned-dispute* represent *SB*s. An *SB* can simply be the parent-child word pair or it can include additional information such as the

relation of child to parent (e.g., *Among-obj-crises*), the part-of-speech (POS) of both child and parent (e.g., *Among/IN-crises/NNS*), the relative order of the two (e.g., *Among-* < *-crises* or *mentioned->-Among*}), or any combination of the above (e.g., *Among/IN-* < *-obj-crises/NNS*).

We define the surface span (*SS*) to be the absolute surface distance between parent and child in an *SB*. For the *SB*s *Among-crises* and *mentioned-Among*, the *SS* values are 5 and 12, respectively. Overall, in the tree in Figure 5.26, there are six *SB*s with *SS* of 1, two *SB*s with *SS* of 2, three *SB*s with *SS* of 3 and one *SB* each for *SS* values 4, 5, 10 and 12.

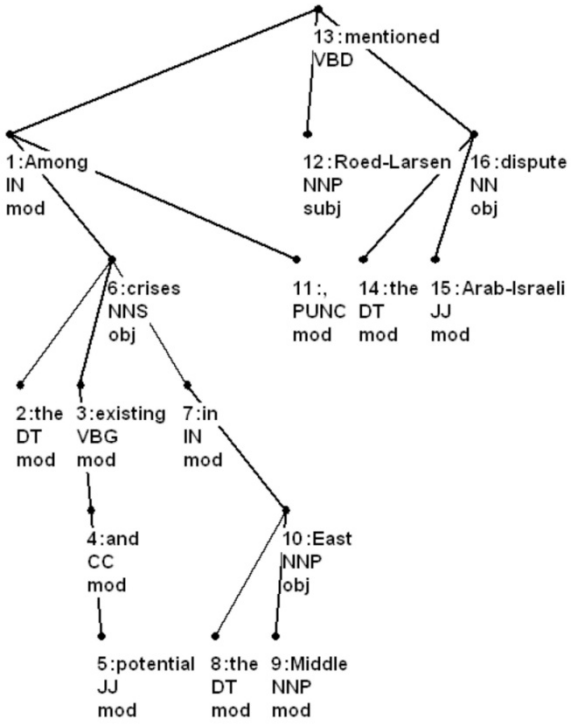


Figure 5.26: A dependency tree analysis for the sentence *Among the existing and potential crises in the Middle East, Roed-Larsen mentioned the Arab-Israeli dispute.*}

For each unique *SS* value, *n*, associated with any *SB* in the hypothesis tree, we define SS_n as the count of all the *SB*s that have an *SS* value of *n*. We also define $SSclip_n$ as the count of all the hypotheses *SB*s (with *SS* value of *n*) that match reference *SB*s. However, if the number of matching hypothesis *SB*s exceeds the maximum seen in any reference tree, we use a partial count equal to (maximum # of reference *SB*s / # of hypothesis *SB*s) in computing $SSclip_n$. This is our variant of *clipping*, used by other precision-based metrics (Papineni 2002) to minimize gaming.

Finally, we define the set *SPANS* to contain all the unique *SS* values seen in the hypothesis tree.

Next, we describe two span-extended *SB* precision sub-scores, which vary in how they use the *SS* of an *SB*: SN_x and *SPN*.

First, the sub-score SN_x is computed as follows:

$$SN_x = \frac{\sum_{n \in SPANS} SSclip_n \times n^x}{\sum_{n \in SPANS} SS_n \times n^x} \quad (5.10)$$

SN_x is basically the span-weighted precision of hypothesis *SB*s matching reference *SB*s. The weighing is controlled through the power term *x*. The default value of *x* is 0, which assigns all *SB*s equal weight regardless of the *SS* value. A power term of 1 effectively multiplies the count of an *SB* by its *SS* value. A multiplier of 2 multiplies the count by the square of the *SS* value (and so on). This allows the user to give a bigger weight to the longer-distance matching spans.

Second, the sub-score SPN is computed as follows:

$$SPN = \frac{1}{|SPANS|} \sum_{n \in SPANS} \frac{SSclip_n}{SS_n} \quad (5.11)$$

SPN is basically the average of all SS-value-specific precision calculations. This scoring approach normalizes the frequency of SS values. This effectively gives more weight to the long-distance SBs because of the Zipfian distribution of SSs: shorter spans appear more frequently than longer spans.

Although the two scoring methods are different, they both give more weight to long-distance dependencies than to short-distance dependencies.

Sub-Score Combination. The segment-level SEPIA score is computed by taking the mean of any subset of the sub-scores described above, including both surface *n*-gram and SB sub-scores. Note that using the surface *n*-gram sub-scores alone is comparable to using BLEU. This serves as a robust back off in case no parse is found. The score is further adjusted by multiplying it with a brevity penalty factor as done in other precision-based metrics (Papineni 2002). Document-level scores are computed as a segment-length-weighted (in words) average of segment scores. Similarly, system-level scores are computed as a document-length-weighted (in segments) average of document scores.

The set of all parameters in the SEPIA package is described in (Habash 2008 SEPIA).

5.6.4.2. SEPIA's Performance in NIST MATR

For the NIST MetricsMATR evaluation, two SEPIA variants that are tuned on different data sets were submitted. SEPIA1 was tuned for MT-06 data, and uses surface unigram, bigram and trigram precision, in addition to the *SPN* subscore over (word, parent, relation, wordPOS, parentPOS and surface relative order). SEPIA2 was tuned for TRANSTAC[24] data, which was quite limited. It only used surface unigrams and bigrams

[24] Spoken Language Communication and Translation System for Tactical Use (TRANSTAC). See http://www.darpa.mil/IPTO/programs/transtac/transtac.asp.

over morphologically de-inflected words. Both SEPIA variants employed length penalty. In terms of speed, SEPIA1 and SEPIA2 ranked 28th and 29th, respectively, out of the 38 NIST MetricsMATR metrics. Both took around 2 hours to finish (compared to \Bleu taking less than 5 minutes but faster than metrics that took over a day). SEPIA1 was the best performer in two conditions: system-level correlation with multi-referenced pairwise and yes/no conditions. Sepia1 was in the top 5 performers in 12 cases out of 45. SEPIA1 always outperformed SEPIA2, which was in top 5 performers in 4 out of 45 conditions. SEPIA1 and SEPIA2 did better with multiple references than single references. They also did better at document and system level than sentence level: neither SEPIA1 nor SEPIA2 was in top 5 performers at sentence-level correlation.

5.6.5. EDPM

5.6.5.1. Introduction

In improving MT metrics, we try to model acceptable variation, whether by modeling word-choice (e.g., METEOR (Banerjee 2005)), by weighting adjacent matches more than non-local matches (e.g. GTM (Turian 2003)) or by modeling syntactic information (Liu 2005; Owczarzak 2007). In keeping to the syntactic approach, we consider a family of Dependency Pair Match (DPM) measures, which follows the labeled-dependency match version of SPARSEVAL (Roark 2006) and the **d/d_var (**Owczarzak 2007) measures. These approaches evaluate hypothesis-reference similarity with an *F* measure over fragments of a labeled dependency structure, which may be generated by a PCFG with deterministic head-finding (Liu 2005; Roark 2006 SParseval) or by extracting the semantic dependencies from an LFG parser ((Cahill *et al*. (2004) in (Owczarzak 2007)). Our specific extension, Expected Dependency Pair Match (EDPM), leverages a publicly available PCFG parser, deterministic head-finding rules, word-level matching and weighted multiple parse alternatives for improved performance.

5.6.5.2. DPM Family of Metrics

DPM is defined as the *F* measure over bags-of-subtrees of the hypothesis translation dependency tree as compared to bags-of-subtrees of the reference translation dependency tree.

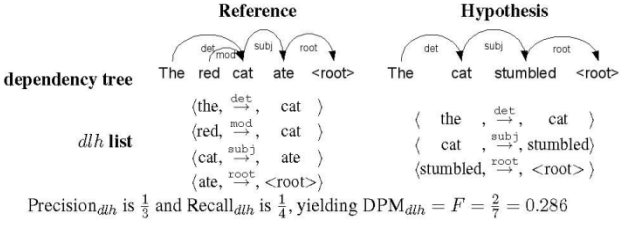


Figure 5.27: Example hypothesis and reference dependency trees and the *dlh* decomposition of each.

Figure 5.27 demonstrates a toy example of the bags-of-dependencies extracted from a hypothesis and reference tree. In the example presented here, we extract only *dlh* dependencies, which are tuples of the following form:

‹Depend, arc-Label, Head›

Different members of the DPM family of metrics may extract different subtrees. We denote the set of extracted tree-components with a trailing DPM_{dlh} extracts all ‹Depend, arc-Label, Head› subtree tuples, roughly equivalent to labeled SPARSEVAL (Roark 2006 Sparseval) and the (Owczarzak 2007) **d** measure. We also consider the $DPM_{1g,2g}$ extractions, which represent unigrams and bigrams, or the $DPM_{dl,lh}$, which extracts all the subtrees ‹Depend, arc-Label› and ‹arc-Label, Head› (roughly equivalent to the (Owczarzak 2007) **d_var** method).

Since the dependency structures of the hypothesis and reference text are hidden, we also explore alternative dependency structures predicted by the parser, to cope with genuine ambiguity (in both hypothesis and reference) and to mitigate the effects of parser error. DPM is well-defined over the n-best list of dependency-structures: when n>1 DPM uses the expectation of bags-of-subtrees rather than the bags-of-subtrees derived from the 1-best parse. In computing the expectation, we first "flatten" the parser probabilities such that $\tilde{p}(x) = \frac{p(x)^\gamma}{\sum_i p(i)^\gamma}$ (where γ is a free parameter) to account for over-confidence in the parser, normalizing so that the resulting probabilities sum to one over the n-best list.

In principle, the DPM family of measures may be implemented with any parser that generates a dependency graph (a single labeled arc for each word, pointing to its head-word). Our reference implementation[25] uses a state-of-the-art PCFG parser (the first stage of Charniak-Johnson, (2005)) to generate a 50-best list of trees for each hypothesis and reference translation, using the parser's default WSJ (Wall Street Journal) training parameters. We use Magerman (1995) head-finding to construct dependency trees, using the Charniak parser's head-finding rules, with three modifications: prepositional and complementizer phrases choose nominal and verbal heads respectively and auxiliary verbs are modifiers of main verbs (rather than the converse).

Arc-labels $d \xrightarrow{A/B} h$ are determined from constituent labels, where the arc label A/B between dependent d and its head h is composed of A (the lowest constituent headed by h and dominating d) and B (the highest constituent headed by d). This strategy is the one adopted in labeled-dependency SPARSEVAL, and it acts as an approximation of the rich semantics generated by (Cahill *et al*. 2004), but with much less knowledge-engineering required. The A/B labels are not as descriptive as LFG semantics, but they have a similar resolution, e.g. the $\xrightarrow{S/NP}$ arc label usually represents a subject dependent of a sentential verb.

[25] http://ssli.ee.washington.edu/people/jgk/dist/edpm/

5.6.5.3. Experimental Validation

Experiments exploring correlation with fluency and adequacy judgments against the LDC Multiple Translation Chinese corpus parts 2 and 4 indicate that the best member of the DPM family uses the full 50-best parses produced by the system, with a "flattening" $\lambda = 0.25$. Using both the partial subtrees *dl,lh* and the string-only statistics *1g,2g* provides an optimal setting of:

$$EDPM = DPM_{1g,2g,dl,lh}, n = 50, \gamma = 0.25$$

This configuration for EDPM has a correlation *r=0.24* against the average fluency and adequacy judgment per-sentence over these corpora.

Measure *s*	All-Arabic	All-Chinese	All
TER	0.51	0.19	0.39
BLEU	-0.40	-0.19	-0.32
EDPM	-0.61	-0.25	-0.47

Table 5.18: Per-document Pearson's *r* of Δ*s* with ΔHter over various measures *s*, examined for each genre in the corpus, for each language in the corpus, and as a whole.

We also explore the EDPM variant's utility on the task of predicting the human-targeted translation edit rate (HTER) on the (unsequestered) GALE evaluation results, and find that per-document differences across systems in EDPM (ΔEDPM) are better correlated with changes in HTER (ΔHTER) than ΔBLEU or ΔTER (Table 5.18).

Note that HTER uses a TER measure to calculate the post-editing work between the hypothesized translation and the human-targeted reference which could, in principle, bias HTER towards a TER measure. EDPM shares no such advantage. EDPM nevertheless has the best correlation of the three measures in both Arabic and Chinese, as well as over the entire corpus.

5.6.5.4. Future Work

Within the DPM framework, there are a number of directions for improving the quality of the scoring function, including increasing the number of parse alternatives, using different parsers, and exploring other (higher order) segmentations of the dependency tree. It would also be useful to assess the sensitivity of the score to parse quality, since very poor quality translations might be difficult to parse.

Alternatively, one might look at combining EDPM with other measures that attempt to account for allowable variation in word choice. Unlike many recently proposed evaluation methods, EDPM does *not* use word-substitution tables or tuned weights (beyond the λ free parameter described above), and yet substantially outperforms BLEU and TER as a predictor of changes to HTER. Perhaps further gains could be achieved by combining the different approaches.

5.6.6. Mulch

5.6.6.1. Introduction

Our work aims to improve automatic MT evaluation by working with the syntactic structure of the reference sentence rather than treating it as a string of words. Our hypothesis is that this will allow a metric to make more sophisticated decisions about the relationship between the system output and the reference sentence, particularly in regard to such aspects as word movement and substitution. We are testing this hypothesis by adapting the METEOR system to use the treebanked representations of reference sentences that are available as part of the Ontonotes project. We refer to this new evaluation approach as MULCH (``Metric Using Large Chunks of Hierarchy'').

METEOR has two basic components. The first is a unigram matching between the system and reference sentences, where the matching is done by exact match and also looser matching using WordNet. The second is a ``fragmentation penalty'' to account for word order, attempting to capture such information as arguments in the wrong order.

We have decomposed the trees for the reference sentences into small tree fragments, representing core pieces of argument structure, with adjuncts and arguments separated. The two components of our adapted METEOR view the reference sentence as a collection of these small tree fragments, with the following impact:

- The unigram matching is able to differentiate between “function words” that can easily differ between a system output and reference translation, and those words that capture a core semantic notion. For example, the complementizer *that* might be missing from a subordinate clause, or the preposition *about* might be used instead of *on* in a prepositional clause. In such cases, by referring to the small tree fragments, we can exclude such function words from the recall calculation.
- METEOR 's fragmentation penalty is not able to distinguish between adjuncts and arguments, although they have very different word order possibilities. We have taken the obvious step, now that we have the trees, of completely rewriting the fragmentation penalty to require arguments to be in the correct order, while adjuncts are free to move.

5.6.6.2. Example

We present here an example sentence from the output of one of the GALE systems, from the Arabic Newswire section of the data. The reference translation for the sentence is (1), and the system output is (2).

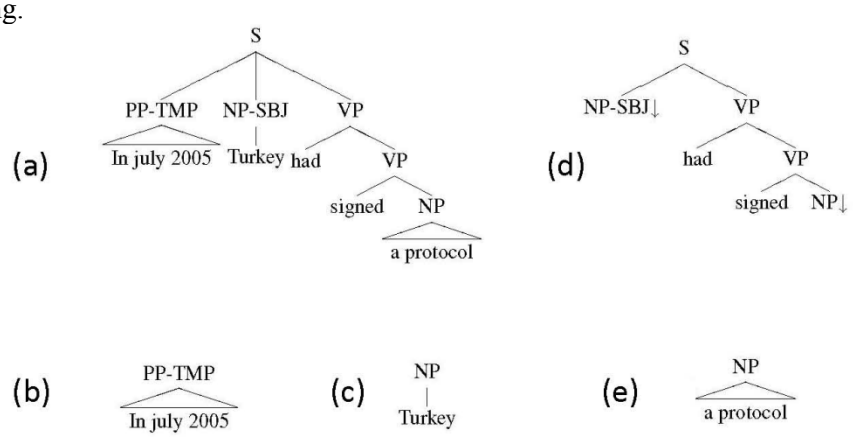
(1) In July 2005 Turkey had signed a protocol ...
(2) Turkey signed in July 2005 a protocol ...

The unigram matching component of METEOR leaves *had* in the reference sentence unmatched, lowering the recall score. In addition, the “fragmentation penalty” lowers the score because the words *In July 2005 Turkey had signed* in the reference sentence do not map to a continuous sequence in the system output, due to the movement of *in July 2005*.

These are clearly not actual errors in the system output though, but rather simply an alternative translation, different from the reference sentence, but just as valid. However, since METEOR only works with words as strings with no linguistic information, it is unable to recognize this and compensate appropriately.

In our approach, we utilize the treebanked representation of the reference string, of which the relevant section is shown below (a). We decompose this tree into smaller tree fragments, associating each one with a lexical item[26]. This results in the four trees in (b), (c) (d) and (e), as well as a ``derivation tree'', not shown here, which encodes how the tree fragments fit together to form the original tree[27].

As mentioned above, we can now improve the unigram matching fmeasure of METEOR by using these tree fragments as a small domain in which the matching can be made loosened. For example, we have encoded that if there is a match with the main verb of a tree, there is no penalty if an auxiliary that is present in the same tree does not map to a word in the system output. In this example, this means that since *signed* in the tree (d) is mapped to *signed* in the system output, there is no penalty for *had* not having a mapping.



Also, while we do not have space to show all the details here, these trees can be used to check that verbs and their arguments are in the same order in the system output as in the reference translation. We have completely rewritten the "fragmentation penalty" aspect of METEOR to take advantage of this new source of information. In this example, the order of *Turkey* and *a protocol* around *signed* is enforced, using the information in tree (d). However, the adjunct in *july 2005* is allowed to appear anywhere[28] and so there is no penalty in our revised METEOR for *in July 2005* appearing in different locations in the reference and system output sentences.

It is worth noting that to some extent our use of the trees is able to simulate what the human annotator does in the edited reference sentences for the HTER evaluation. For example, for this particular sentence the edited reference sentence is edited:

[26] This is very much inspired by the long line of work in Tree Adjoining Grammar Joshi and Schabes (1997).

[27] The ↓ indicates a node where another tree must be substituted into it, such as the tree for Turkey in (c) into the NP-SBJ node in (d).

[28] This is of course a simplification, since adjuncts do not have complete freedom of movement.

(3) Turkey signed in July 2005 a protocol ...

What the human annotator has done is adjust for the missing *had* and the movement of *in July 2005*, just as we have been able to do automatically by making use of the linguistic information encoded in the trees.

5.6.6.3. Evaluation

To test the modified METEOR, we calculated the following three scores for a team's output, for each sentence:

- HTER: using tercom on the system output and human-edited reference sentence.
- Original METEOR: using the system output and the (unedited) reference sentence.
- Modified METEOR: using the same data as for the previous, along with the treebanked representation of the reference sentence.

We then calculate the Pearson correlation between (HTER, original METEOR) and (HTER, modified METEOR).

Focusing for now on the Arabic Newswire part of the data, we obtain the scores for a development slice of the data, showing the increase in correlation from (HTER, original METEOR) to (HTER, modified METEOR):

AGILE:	0.66 → 0.69
ROSETTA:	0.65 → 0.66
NIGHTINGALE:	0.62 → 0.63

While there is some improvement, it is clearly not enough. Our hypothesis, based on some examination of the data, is that while our modified METEOR is able to improve the correlation with HTER when the system output and reference translation are relatively close, it has very little to work with if the system output and reference translation are far apart, since in that case the edited reference sentence used for HTER is so greatly different from the original reference that our modified METEOR is not able to simulate the construction of the edited reference sentence.

That is, we view the construction of the HTER sentence as consisting of four aspects:

- deletions/insertions: METEOR tracks this, while our modified version can improve this.
- synonyms: METEOR tracks this.
- word order movement: METEOR tracks this, while our modified version can improve this.
- severe paraphrases substantially altering the reference sentence. Neither METEOR nor our modified version can track this.

Our plan is to measure the distance between the original and edited references, in effect measuring ``severe paraphrases'', and determine if there is a threshold of distance such that for distances less than that threshold our modified METEOR shows a substantial improvement in correlation compared with the original METEOR.

5.6.7. The Stanford RTE-based Metrics (RTE and RTE+MT)

5.6.7.1. Introduction

The first generation of metrics developed for the automatic evaluation of machine translation output (such as BLEU (Papineni *et al*. 2002) and NIST (Doddington 2002)) were based on surface overlap, looking for identical word sequences between a system translation and one or more reference translations. However, recent work such as Callison-Burch *et al*. (2006) has identified serious problems with these metrics, such as unreliability at the level of individual segments, a bias towards statistical phrase-based MT systems, and (at least for some language pairs) low correlations between scores and human judgments. This casts a shadow on the frequent use of higher BLEU scores as a necessary and sufficient indicator of improvement in MT.

More recent metrics address these shortcomings by refining the matching problem, for example by phrasing it as an edit sequence (Ter, Snover *et al*. 2006), or by integrating synonymy information from WordNet (Meteor, Banerjee and Lavie 2005). Our proposal is to go one step further and determine the adequacy of a system hypothesis with respect to a reference translation completely on a semantic level that generalizes over the different kinds of variability of linguistic realization that can arise through syntactic rearrangements (such as active/passive diatheses or adverb placement) or lexical and paraphrastic variation. To do so, we exploit the considerable similarities between the MT evaluation task and the task of Recognition of Textual Entailment or RTE (Dagan *et al*. 2006). Given two short segments of text (premise and hypothesis), the recognition of textual entailment decides whether the hypothesis is entailed by the premise or not. Knowledge about the likelihood of textual entailment has been found to be beneficial for a range of applications, e.g., Word Sense Disambiguation or answer validation in Question Answering (Harabagiu and Hickl 2006).

Transferred to the domain of MT evaluation, our intuition is that there a good translation should be equivalent in meaning to the reference translation, Hence the likelihood of mutual entailment between system output and reference should be high for good translations, and low for bad translations.

However, there are also evident differences between RTE and MT evaluation, both in terms of the data (entailment usually assumes that sentences are well-formed, which is often not true in MT), and in terms of the task (entailment is a “stricter”, and asymmetrical, task). Thus, we advocate a modular approach that predicts MT quality on the basis of entailment features, but can adjust the importance of these features for the new task. The resulting entailment-inspired approach has the potential of addressing the problems described in the first paragraph. First, entailment decisions are naturally made at the segment level. Second, textual entailment provides a framework for the integration of different kinds of linguistic analysis techniques into MT evaluation. Finally, a deeper

linguistic analysis can reduce biases towards particular system architectures.

5.6.7.2. Metric Implementation

The entailment features we use are computed by the Stanford RTE system (MacCartney *et al*. 2006). The input to the system is a pair of a premise and a hypothesis. We first perform a linguistic analysis whose result is a typed dependency graph. Next, an alignment is constructed between words of either sentence as the alignment that scores highest according to lexical similarity scores from about ten semantic resources and syntactic parallelism features. In the third phase, we construct features which model a range of syntactic, lexical, and semantic phenomena, such as factivity, polarity, monotonicity, matches and mismatches of named entities, and the quality of the alignment. The final score for each segment pair is computed by a regression model over the set of these entailment features. In contrast to the general entailment task, the features are computed for both directions to make a prediction for MT evaluation. This captures that system output and reference should entail each other.

We compare this system (RTE) against two individual MT metrics (BLEU, TER) and a regression model over different parameterizations of the BLEU, NIST, and TER metrics (MT). Finally, we combine the features of the MT regression model with the features of the RTE regression model to obtain a hybrid model (RTE+MT). We estimate the weights of all models from annotated translation corpora.

	BLEU	TER	MT	RTE	RTE+MT
NIST	60.0	64.0	65.(	63	68.3
WMT	35.9	37.5	39.(	42	45.7

Table 5.19: Correlation between human judgments and predictions on two corpora (Spearman's ρ)

5.6.7.3. Experiment 1

We first test the metric on two publicly available corpora with human judgments. The first one, NIST, includes data from the Open MT Evaluations of 2006 and 2008. The second one, WMT, consists of the 2006 and 2007 datasets of the ACL/NAACL WMT workshops. We used ten-fold cross-validation for (regularized) parameter estimation and testing. Table 5.19 shows Spearman's ρ correlation coefficients between the models' predictions and human judgments. Results for the two corpora differ in absolute values, due, e.g., to the use of a 5-point adequacy scale (WMT) vs. a 7-point scale (NIST). However, the main tendencies are the same: MT and RTE outperform individual metrics, but are not clearly ordered. The hybrid system (RTE+MT) substantially outperforms all other systems. Apparently, MT evaluation benefits from the complementary types of information it obtains from shallow and entailment features.

5.6.7.4. Experiment 2

We submitted the RTE+MT model from Experiment 1, with feature weights estimated on the union of the NIST and WMT corpora, to the NIST MetricsMATR'08 single-reference track. The evaluation dataset was news and web data drawn from the NIST MT Eval and GALE programs, with some Transtac dialogue data. Across all performance analyses, our system was generally among the upper half. In the prediction of 7-point adequacy scores, RTE+MT obtained rank 14th of 39, with a correlation of $\rho = 0.61$ (best system: 0.68). In the prediction of qualitative scores, it attained 7th place ($\rho = 0.51$, vs. 0.57 for the best system). It was however the best system for the prediction of HTER scores ($\rho = -0.56$), the main GALE evaluation target. We hypothesize that the inferior performance in MetricsMATR compared to Exp. 1 is due to the following factors: (a), overfitting on training data due to a large number of parameters; (b), difficulty of structural features to deal with the often ungrammatical speech transcriptions in the Metric-sMATR data; (c), the absence of score normalization at the document or dataset level.

5.6.7.5. Conclusion

Our results show that the use of textual entailment features for MT evaluation is a promising perspective. We found that a vanilla RTE system— originally developed for well-formed English text—was mostly robust enough to compete with state-of-the-art MT evaluation metrics without adaptation. In particular, we found entailment-based features can be combined with traditional MT evaluation metrics to obtain a hybrid "best of both words" system that consistently outperforms either individual approach, which allows for the incremental improvement of evaluation schemes.

Our current metric still has a number of limitations which are the subject of on-going research. Currently, a large hurdle to its practical use is its high resource requirements. Our metric requires several seconds to score one segment pair. While this is fast enough to score large corpora for evaluation purposes, it is clearly too slow for integration into minimum error rate training (MERT). However, a number of simple changes can reduce runtime considerably. In addition, the independence between individual segment pairs makes scoring easily parallelizable.

References

Agarwal, A. and A. Lavie. 2008. Meteor, M-BLEU and M-TER: Evaluation metrics for high-correlation with human rankings of machine translation output. *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 115–118, Columbus, Ohio, June. Association for Computational Linguistics.

ALPAC (Automatic Language Processing Advisory Committee). 1966. Report of the ALPAC; Language and Machines: Computers in Translation and Linguistics. Division of Behavioral Sciences, National Academy of Sciences, National Research Council, Washington, D.C.

Banerjee, S. and A. Lavie. 2005. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. *Proceedings of the ACL 2005 Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization*, pages 65–72, Ann Arbor, Michigan, June.

Bannard, C. and C. Callison-Burch. 2005. Paraphrasing with Bilingual Parallel Corpora. *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics* (ACL 2005), pages 597–604, Ann Arbor, Michigan, June.

Belvin, R.S., S. Rieheman and K. Precoda. 2004. A Fine-Grained Evaluation Method for Speech-to-Speech Machine Translation Using Concept Annotations. *Proceedings of the Fourth International Conference On Language Resources and Evaluation* (LREC 2004), Lisbon, Portugal.

Blatz, J., E. Fitzgerald, G. Foster, S. Gandrabur, C. Goutte, A. Kulesza, A. Sanchis and N. Ueffing. 2003. Confidence Estimation for Machine Translation. *Technical Report Natural Language Engineering Workshop Final Report*, Johns Hopkins University.

Cahill, A., M. Burke, R. O'Donovan, J. van Genabith and A. Way. 2004. Long-distance dependency resolution in automatically acquired wide-coverage PCFG-based LFG approximations. *Proc. ACL*, pages 319–326.

Callison-Burch, C., M. Osborne and P. Koehn. 2006. Re-Evaluating the Role of Bleu in Machine Translation Research. *Proceedings of EACL-2006*, pages 249–256, Trento, Italy.

Callison-Burch, C., C. Fordyce, P. Koehn, C. Monz and J. Schroeder. 2007. (meta-) evaluation of machine translation. *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 136–158, Prague, Czech Republic, June. Association for Computational Linguistics.

Charniak, E. and M. Johnson. 2005. Coarse-to-fine n-best parsing and maxent discriminative reranking. *Proceedings of the 43rd Annual Meeting of the*

Association for Computational Linguistics (ACL'05), pages 173–180, Ann Arbor, Michigan, June. Association for Computational Linguistics.

Child, J.R., T. Ray and P. Lowe, Jr. 1993. Proficiency and Performance in Language Testing. *Applied Language Learning*, Vol. 4.

Church, K. and E. Hovy. 1993. Good applications for crummy machine translation. *Machine Translation*, 8:239–258.

Coughlin, D. 2003. Correlating Automated and Human Assessments of Machine Translation Quality. *Proceedings of MT Summit IX*, pages 63–70, New Orleans, LA.

Dagan, I., O. Glickman and B. Magnini. 2006. The PASCAL Recognising Textual Entailment Challenge. Quiñonero-Candela, J., Dagan, I., Magnini, B., d'Alché-Buc, F. (Eds.) *Machine Learning Challenges. Lecture Notes in Computer Science*, Vol. 3944, pp. 177-190, Springer.

Doddington, G. 2002. Automatic Evaluation of Machine Translation Quality Using N-gram Co-Occurence Statistics. *Proceedings of the Human Language Technology* (Notebook), pages 128–132, San Diego, CA.

Fellbaum, C. 1998. WordNet: An Electronic Lexical Database. MIT Press. http://www.cogsci.princeton.edu/˜wn [2000, September 7].

Fisher, F. and C.R. Voss. 1997. Falcon, an mt system support tool for nonlinguists. *Proceedings of the Advanced Information Processing and Analysis Conference* (AIPA 97), pages 182–191, McLean, VA.

Fisher, F., C. Schlesiger, L. Decrozant, R. Zuba, M. Holland and C.R. Voss. 1999. Searching and translating arabic documents on a mobile platform. *Proceedings of the Advanced Information Processing and Analysis Conference* (AIPA 99), Washington, DC.

Frederking, R. and S. Nirenburg. 1994. Three Heads are Better than One. *Proceedings of the Fourth Conference on Applied Natural Language Processing* (ANLP-94), Stuttgart, Germany.

Garofolo, J.S., T. Robinson, and J. G. Fiscus. 1994. The development of file formats for very large speech corpora: Sphere and shorten. *Proceeding of ICASSP*.

Gates, D., A. Lavie, L. Levin, A. Waibel, M. Gavald`a, L. Mayfield, M. Woszczyna and P. Zhan. 1996. End-to-End Evaluation in JANUS: a Speech-to-Speech Translation System. *Proceedings of the European Conference on Artificial Intelligence* (ECAI-1996) (Workshop on "Dialogue Processing in Spoken Language"), Budapest, Hungary, August.

Gimenez, J. and L. Marquez. 2007. Linguistic features for automatic evaluation of heterogenous MT systems. *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 256–264, Prague, Czech Republic, June. Association for Computational Linguistics.

Habash, N. and A. Elkholy. 2008. Sepia: Surface span extension to syntactic dependency precision-based mt evaluation. *Proceedings of the NIST Metrics for Machine Translation Workshop at the Association for Machine Translation in the Americas conference*, AMTA-2008, Waikiki, Hawaii.

Hutchins, W.J. 2001. Machine Translation Over Fifty Years. *Histoire Épistémologie Langage*, I(23):7–31. 2008. ILR language skill level descriptions. http://www.govtilr.org.

Jones, D. and W. Shen. 2006. Two New Experiments for ILR-Based MT Evaluation. *Proceedings of Association for Machine Translation in the Americas*.

Jones, E., T. Oliphant, P. Peterson, *et al*. 2001–. *SciPy*: Open source scientific tools for Python.

Jones, D.A., W. Shen, N. Granoien, M. Herzog and C. Weinstein. 2005. Measuring translation quality by testing English speakers with a new defense language proficiency test for Arabic. *Proceedings of 2005 International Conference on Intelligence Analysis*, May 2-6, 2005, McLean, VA., May.

Jones, D., M. Herzog, H. Ibrahim, A. Jairam, W. Shen, E. Gibson and M. Emonts. 2007. ILR-based MT comprehension test with multi-level questions. *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics*; Companion Volume, Short Papers, pages 77–80, Rochester, New York, April. Association for Computational Linguistics.

Joshi, A.K. and Y. Schabes. 1997. Tree-adjoining grammars. *G. Rozenberg and A. Salomaa, eds., Handbook of Formal Languages*, Volume 3: Beyond Words, pages 69–124. Springer, New York.

Kauchak, D. and R. Barzilay. 2006. Paraphrasing for Automatic Evaluation. *Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL*, pages 455–462.

King, M. 1996. Evaluating Natural Language Processing Systems. *Communication of the ACM*, 29(1):73–79, January.

Knight, K. and I. Chander. 1994. Automated Postediting of Documents. *Proceedings of National Conference on Artificial Intelligence (AAAI)*, pages 779–784, Seattle, Washington.

Laoudi, J., C. Tate and C.R.Voss. 2006. Task-based mt evaluation: From who/when/where extraction to event understanding. *Proceedings of the Fifth International Conference on Language Resources and Evaluation*, LREC-2006, pages 2048–2053, Genoa, Italy.

Lavie, A. and A. Agarwal. 2007. METEOR: An automatic metric for MT evaluation with high levels of correlation with human judgments. *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 228–231, Prague, Czech Republic, June. Association for Computational Linguistics.

Lavie, A., K. Sagae and S. Jayaraman. 2004. The Significance of Recall in Automatic Metrics for MT Evaluation. *Proceedings of the 6th Conference of the Association for Machine Translation in the Americas* (AMTA-2004), pages 134–143, Washington, DC, September.

Leusch, G., N. Ueffing and H. Ney. 2006. CDER: Efficient MT Evaluation Using Block Movements. *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics* (EACL 2006).

Levenshtein, V. I. 1966. Binary Codes Capable of Correcting Deletions, Insertions, and Reversals. *Soviet Physics Doklady*, 10:707–710.

Levin, L., D. Gates, A. Lavie, F. Pianesi, D. Wallace, T. Watanabe and M. Woszczyna. 2000. Evaluation of a Practical Interlingua for Task-Oriented Dialogue. *Proceedings of ANLP/NAACL-2000 Workshop on Applied Interlinguas*, pages 18–23, Seattle, WA.

Liu D. and D. Gildea. 2005. Syntactic features for evaluation of machine translation. *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization,* pages 25–32, Ann Arbor, Michigan.

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

Madnani, N., N.F. Ayan, P. Resnik and B.J. Dorr. 2007. Using paraphrases for parameter tuning in statistical machine translation. *Proceedings of the Workshop on Statistical Machine Translation*, Prague, Czech Republic, June. Association for Computational Linguistics.

Madnani, N., P. Resnik, B.J. Dorr and R. Schwartz. 2008. Are Multiple Reference Translations Necessary? Investigating the Value of Paraphrased Reference Translations in Parameter Optimization. *Proceedings of the Eighth Conference of the Association for Machine Translation in the Americas*, October.

Magerman, D.M. 1995. Statistical decision-tree models for parsing. *Proc. ACL*, pages 276–283.

Matusov, E., G. Leusch, O. Bender and H. Ney. 2005. Evaluating machine translation output with automatic sentence segmentation. *Proceedings of the International Workshop on Spoken Language Translation* (IWSLT) 2005.

Moore, R. C. and C. Quirk. 2008. Random restarts in minimum error rate training for statistical machine translation. *Proceedings of the 22nd International Conference on Computational Linguistics* (Coling 2008), pages 585–592, August.

Nelder, J. A. and R. Mead. 1965. A simplex method for function minimization. *Computer Journal*, 7:308–313.

Nießen, S., F.J. Och, G. Leusch, and H. Ney. 2000. An evaluation tool for machine translation: Fast evaluation for MT research. *In Proceedings of the 2nd International Conference on Language Resources and Evaluation* (LREC-2000), pages 39–45.

NIST and LDC. 2007. Post Editing Guidelines for GALE Machine Translation Evaluation, Version 3.0.2, May 25.

Nübel, Rita. 1997. End-to-End Evaluation in VERBMOBIL I. *Proceedings of MT Summit VI*, pages 232–239, San Diego, CA.

Och, F.J. and H. Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics* (ACL), pages 295–302, Philadelphia, PA, July.

Och, F.J.. 2003. Minimum Error Rate Training in Statistical Machine Translation. *Proceedings of the 41st Annual Meeting of the Association of Computational Linguistics*, pages 160–167, July.

Owczarzak, K., J. van Genabith and A. Way. 2007. Labeled dependencies in machine translation evaluation. *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 104–111, Prague, Czech Republic. Association for Computational Linguistics.

Papineni, K., S. Roukos, T. Ward and W. Zhu. 2002. BLEU: a Method for Automatic Evaluation of Machine Translation. *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, PA.

Porter, M.F. 1980. An algorithm for suffic stripping. *Program*, 14(3):130–137.

Powell, M. J. D. 1964. An efficient method for finding the minimum of a function of several variables without calculating derivatives. *Computer Journal*, 7:155–162.

Przybocki, M., K. Peterson and S. Bronsart. 2008. Official results of the NIST 2008. *"Metrics for MAchine TRanslation" Challenge (Metrics-MATR08)*. http://nist.gov/speech/tests/metricsmatr/2008/results/, October.

Roark, B., M. Harper, E. Charniak, B. Dorr, M. Johnson, J.G. Kahn, Y. Liu, M. Ostendorf, J. Hale, A. Krasnyanskaya, M. Lease, I. Shafran, M. Snover, R. Stewart and L. Yung. 2006. SParseval: Evaluation metrics for parsing speech. *Proc. LREC*.

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

Russo-Lassner, G., J. Lin and P. Resnik. 2005. A Paraphrase-Based Approach to Machine Translation Evaluation. *Technical Report LAMPTR-125/CS-TR-4754/UMIACS-TR-2005-57*, University of Maryland, College Park.

Sanders, G.A., S. Bronsart, S. Condon and C. Schlenoff. 2008. Odds of Successful Transfer of Low-level Concepts: A Key Metric for Bidirectional Speech-to-speech Machine Translation in DARPA's TRANSTAC Program. *Proceedings of the Sixth International Conference on Language Resources and Evaluation* (LREC-2008), Marrakesh, Morocco, May. European Language Resources Association (ELRA).

Snover, M., B. Dorr and R. Schwartz. 2004. A Lexically-Driven Algorithm for Disfluency Detection. *Proceedings of HLT/NAACL*, pages 157–160.

Snover, M., B. Dorr, R. Schwartz, L. Micciulla and J. Makhoul. 2006. A Study of Translation Edit Rate with Targeted Human Annotation. *Proceedings of Association for Machine Translation in the Americas* (AMTA-2006), pages 223–231, Cambridge, Massachusetts.

Tate, C.R. and C.R. Voss. 2006. Combining evaluation metrics via loss functions. *Proceedings of the Association for Machine Translation in the Americas conference*, AMTA-2006, Boston, MA.

Tate, C.R. 2007. An Investigation of the Relationship between Automated Machine Translation Evaluation Metrics and User Performance on an Information Extraction Task. *Ph.D. thesis, University of Maryland,* College Park, MD.

Tate, C.R. 2008. A statistical analysis of automated mt evaluation metrics for assessments in task-based mt evaluation. *Proceedings of the Association for Machine Translation in the Americas conference*, AMTA-2008, pages 182–191, Waikiki, Hawaii.

Taylor, K. and J. White. 1998. Predicting what mt is good for: User judgments and task performance. *Proceedings of the Association for Machine Translation in the Americas conference*, AMTA-1998, Langhorne, PA.

Tillmann, C., S. Vogel, H. Ney, A. Zubiag and H. Sawaf. 1997. Accelerated DP Based Search For Statistical Translation. *European Conference on Speech Communication and Technology*, pages 2667–2670, Rhodes, Greece, September.

Turian, J.P., L. Shen and D.I. Melamed. 2003. Evaluation of machine translation and its evaluation. *Proc. MT Summit IX*, pages 386–393, New Orleans, LA.

Vanni, M., C.R. Voss and C.R. Tate. 2004. Ground truth, reference truth and "omniscient truth"—parallel phrases in parallel texts for mt evaluation. *Proceedings of the Fourth International Conference On Language Resources and Evaluation* (LREC 2004), pages 10–13, Lisbon, Portugal.

Voss, C.R. and C.R. Tate. 2006. Task-Based Evaluation of Machine Translation (MT) Engines: Measuring How Well People Extract Who, When, Where Type Elements in MT Output. *Proceedings of the 11th Annual Conference of the European Association for Machine Translation* (EAMT-2006), pages 203–212, Oslo, Norway.

White, J.S. and T. O'Connell. 1994. Evaluation in the ARPA Machine Translation Program: 1993 Methodology. *Proceedings of the ARPA HLT Workshop*, Plainsboro, NJ.

White, J.S., T. O'Connell and F. O'Mara. 1994. The ARPA MT Evaluation Methodologies: Evolution, Lessons, and Future Approaches. *Proceedings of the First Conference of the Association for Machine Translation*, pages 193–205.

White, J.S., J.B. Doyon and S.W. Talbott. 2000. Task Tolerance of MT Output in Integrated Text Processes. *ANLP/NAACL 2000: Embedded Machine Translation Systems*, pages 9–16.

Wilks, Y. 2008. *Machine Translation: Its Scope and Limits*. Springer Verlag, New York, NY.

Zhou, L., C.Y. Lin and E. Hovy. 2006. Re-evaluating Machine Translation Results with Paraphrase Support. *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing* (EMNLP 2006), pages 77–84.