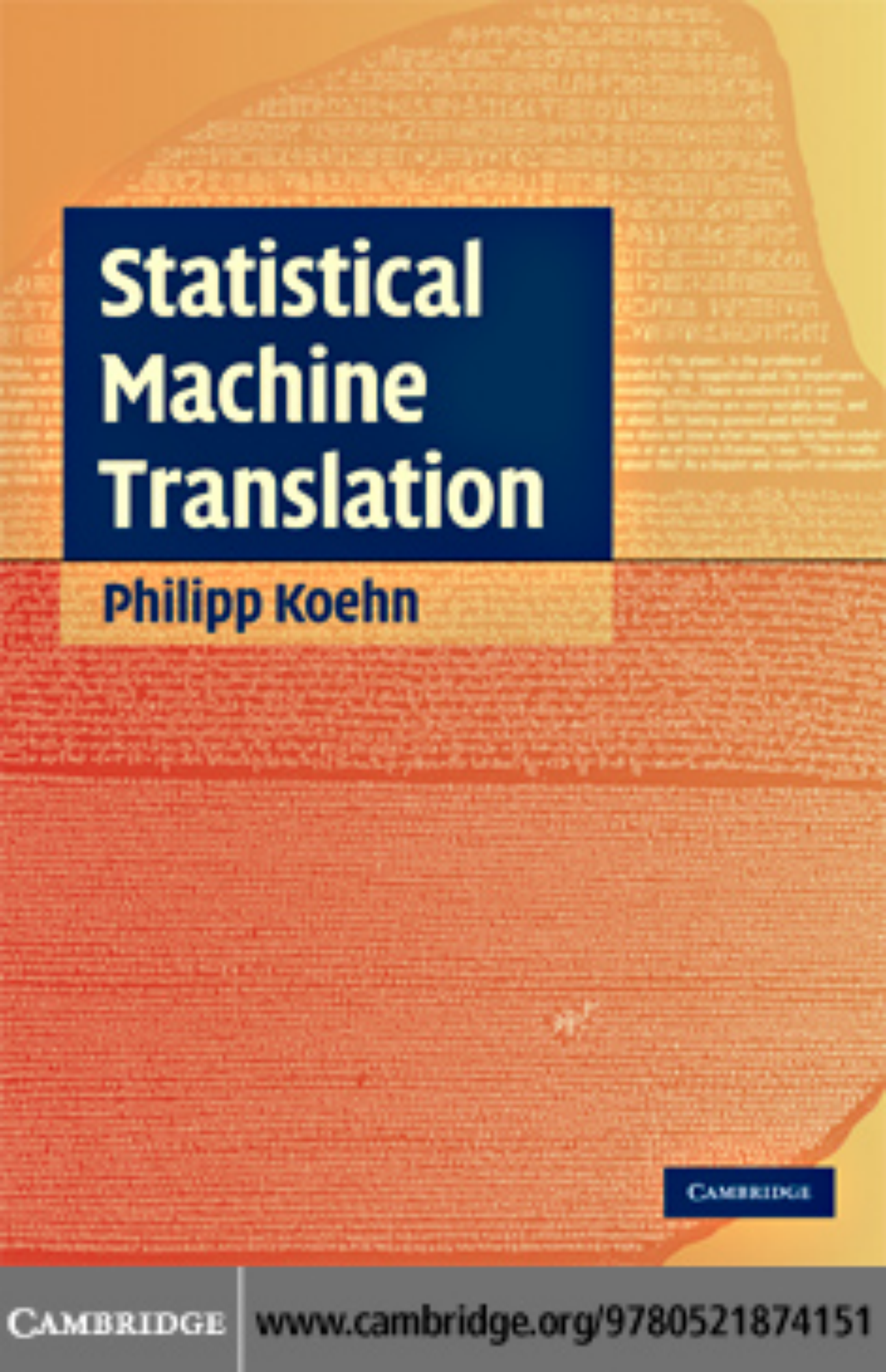
Statistical
Machine
Translation
Philipp Koehn
CAMBRIDGE
CAMBRIDGE
www.cambridge.org/9780521874151

Chapter 8
Evaluation

How good are statistical machine translation systems today? This simple question is very hard to answer. In contrast to other natural language tasks, such as speech recognition, there is no single right answer that we can expect a machine translation system to match. If you ask several different translators to translate one sentence, you will receive several different answers.

Figure 8.1 illustrates this quite clearly for a short Chinese sentence. All ten translators came up with different translations for the sentence. This example from a 2001 NIST evaluation set is typical: translators almost never agree on a translation, even for a short sentence.

So how should we evaluate machine translation quality? We may ask human annotators to judge the quality of translations. Or, we may compare the similarity of the output of a machine translation system with translations generated by human translators. But ultimately, machine translation is not an end in itself. So, we may want to consider how much machine-translated output helps people to accomplish a task, e.g., get the salient information from a foreign-language text, or post-edit machine translation output for publication.

This chapter presents a variety of evaluation methods that have been used in the machine translation community. **Machine translation evaluation** is currently a very active field of research, and a hotly debated issue.

machine translation evaluation

Figure 8.1 Ten different translations of the same Chinese sentence (a typical example from the 2001 NIST evaluation set).

这个 机场 的 安全 工作 由 以色列 方面 负责 .
Israeli officials are responsible for airport security.
Israel is in charge of the security at this airport.
The security work for this airport is the responsibility of the Israel government.
Israeli side was in charge of the security of this airport.
Israel is responsible for the airport's security.
Israel is responsible for safety work at this airport.
Israel presides over the security of the airport.
Israel took charge of the airport security.
The safety of this airport is taken charge of by Israel.
This airport's security is the responsibility of the Israeli security officials.

8.1 Manual Evaluation

An obvious method for evaluating machine translation output is to look at the output and judge by hand whether it is correct or not. Bilingual evaluators who understand both the input and output language are best qualified to make this judgment. Such bilingual evaluators are not always available, so we often have to resort to monolingual evaluators who understand only the target language but are able to judge system output when given a **reference translation**.

reference translation

Typically, such evaluation is done sentence by sentence, but a longer document context may be essential to carry out the judgments. For instance, the resolution of pronouns may only be faithfully evaluated in context.

8.1.1 Fluency and Adequacy

Manual evaluations using a harsh correctness standard – is the translation perfect or not? – have been done, although usually only on short sentences, where there is a reasonable chance that no mistakes are made by the machine translation system. A more common approach is to use a graded scale when eliciting judgments from the human evaluators.

Moreover, correctness may be too broad a measure. It is therefore more common to use the two criteria fluency and adequacy:

fluency

Fluency: Is the output good fluent English? This involves both grammatical correctness and idiomatic word choices.

adequacy

Adequacy: Does the output convey the same meaning as the input sentence? Is part of the message lost, added, or distorted?

See Figure 8.2 for an example from an evaluation tool that elicits fluency and adequacy judgments from a human annotator. The annotator is given the following definitions of adequacy and fluency:

Adequacy	
5	all meaning
4	most meaning
3	much meaning
2	little meaning
1	none

Fluency	
5	flawless English
4	good English
3	non-native English
2	disfluent English
1	incomprehensible

These definitions are very vague, and it is difficult for evaluators to be consistent in their application. Also, some evaluators may generally be more lenient when assigning scores (say, giving an average of 4) than others (say, giving an average of 2).

For example, in the evaluation presented by Koehn and Monz [2005] the average fluency judgement per judge ranged from 2.33 to 3.67, the average adequacy judgement per judge ranged from 2.56 to 4.13. See also Figure 8.3 for the judgments given out by the five most prolific judges (over 1000 judgments each).

normalizing evaluations

We would therefore like to **normalize** the judgments. Ideally, all evaluators use scores around the same average. The average $\bar{x}$ of a set of judgments $\{x_1, ..., x_n\}$ is defined as

$$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i \tag{8.1}$$

If we want all judges to have the same average, say 3, then we need to adjust the individual scores x_i by adding in an adjustment value $3 - \bar{x}$. We may also adjust the variance of scores in the same way.

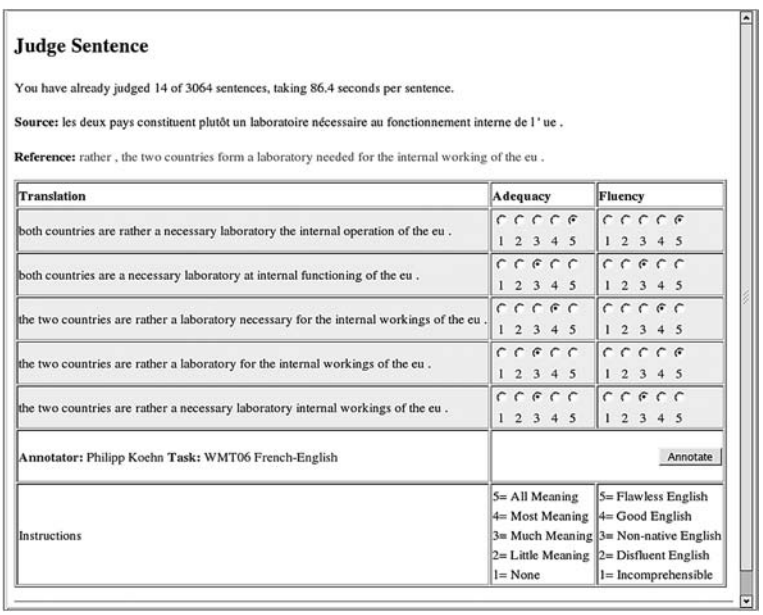

Figure 8.2 Evaluation tool to elicit judgments of translation quality: Five different system outputs are presented, to be scored on a scale of 1–5 for fluency (good English) and adequacy (correct meaning).

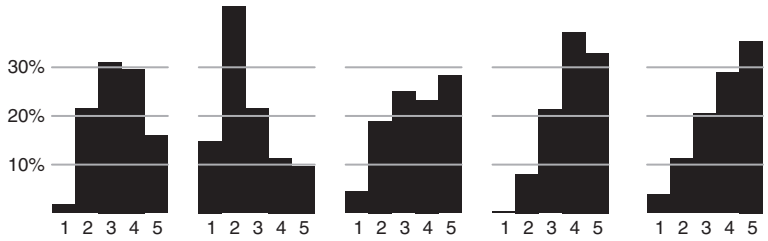

Figure 8.3 Histograms of adequacy judgments by different human evaluators in the WMT 2006 evaluation: Different evaluators use the scale 1–5 in remarkably different ways. The same is true for fluency judgments.

Judging adequacy is tricky. The human mind is quite adept at filling in missing information. Consider the following: If you first read the system output, you may be very puzzled by its meaning, which only becomes clear after reading the reference translation (or input sentence). But if you first read the reference or input sentence and then read the system output, you may not notice that the system output is very garbled and may come to the conclusion that the gist of the meaning can be found in there. The latter may also be the case if you have sufficient domain knowledge that helps you understand the meaning the sentence.

Recent evaluation campaigns have shown that judgments of fluency and adequacy are closely related. This may be not completely surprising, since a sentence in garbled English typically also carries less meaning. But this may also point to the difficulty that humans have in distinguishing the two criteria.

ranking translations

Instead of judging fluency and adequacy on an absolute scale, it is typically easier to **rank** two or more systems against each other on a sentence-by-sentence basis. In the case of two systems, the question *Is system output A better than system output B, or worse, or indistinguishable?* is typically answered by human evaluators in a more consistent manner than questions about adequacy or fluency.

8.1.2 Goals for Evaluation

Before we move on to other evaluation metrics, let us review what we expect from such a metric.

low-cost metric

From a practical point of view, a metric should have **low cost**; i.e., it should be possible to quickly and cheaply carry out evaluations of a new system, of a new domain, etc. Cost is the major disadvantage of evaluation metrics that include human evaluators, especially bilingual evaluators. Cost may be measured in time or money spent on the evaluation.

tunable metric

Fully automatic metrics may be **tunable**, i.e., directly used in the automatic system optimization.

meaningful metric

For an evaluation metric to rank systems against each other is useful, but ideally we would like to have a **meaningful** metric. Recall that

the leading question of this chapter was *How good is statistical machine translation today?* Does an adequacy score of 3.5 really answer that, or does it say more about the leniency of the evaluator?

consistent metric

Moreover, we want an evaluation metric to be **consistent**. Consistency should be maintained across many dimensions. Different evaluators using the same metric should come to the same conclusions. This is called **inter-annotator agreement**. But we would also like the evaluation on one part of the test corpus to be consistent with the evaluation on another part. If there is high fluctuation, i.e., the metric is not **stable**, this means that we need large test corpora to ensure that the results are reliable.

inter-annotator agreement

stable metric

correct judgment

Finally, we want an evaluation metric to come up with the **correct** result. Here, unfortunately we are missing one essential element. We can compare outcomes from one evaluation metric against another, but there is no real ground truth. Since fluency and adequacy judgments are currently seen as the metrics that are closest to ground truth, new metrics are generally assessed by how much they correlate with these judgments.

8.1.3 Other Evaluation Criteria

In this chapter, we are primarily concerned with evaluation metrics that reflect how good the machine translation output quality is. For the practical deployment of machine translation systems in an operational environment, several other issues are important.

system speed

One issue is **speed**. We as researchers are somewhat concerned with this issue, since we would like our experiments to finish quickly, so we can examine their results. Typical statistical machine translation research systems have speeds of about 10–100 words per second, but this may not be fast enough for practical deployment. Of course, there is a trade-off between speed and quality: Higher translation speeds may be obtained with some loss of quality, for instance by more pruning in decoding that leads to more search errors.

system size

The **size** of the system plays an important role, even for research systems, which have to run on the available machines. Use of machine translation systems in the field, e.g., on hand-held devices, carries with it tighter constraints on system size.

integration into workflow

Other issues come from the **integration** of a machine translation system into the workflow of an application environment. Does the system interface well with other applications? Is it easy to use? Is it reliable (does it crash)? In this book, we are less concerned with these questions, but they do play an important role in the usefulness of machine

translation systems. Often, machine translation is not used, simply because it is too complicated.

When deploying machine translation systems in a specific environment to translate documents from a specific domain, the system's support for **domain adaptation** and **customization** plays a major role. Users prefer a system to act predictably and may want to correct common errors.

domain adaptation
customization

8.2 Automatic Evaluation

When it comes to evaluating machine translation systems, we tend to put most trust into the judgment of human evaluators who look at the output of several systems, examine it sentence by sentence, assess each sentence, and conclude with an overall score for each system.

This method, however, has one major disadvantage. It takes a lot of time, and if the evaluators expect to be paid, also a lot of money. The typical statistical machine translation researcher, on the other hand, has no money and would like to carry out evaluations very frequently, often many per day to examine different system configurations.

Therefore, we prefer to have an automatic method for assessing the quality of machine translation output. Ideally, we would like a computer program to tell us quickly whether our system got better after a change, or not. This is the objective of **automatic machine translation evaluation**.

automatic evaluation

Much progress has recently been made in this field, to the point that machine translation researchers trust automatic evaluation metrics and design their systems based on the rise and fall of automatic evaluation scores. However, automatic evaluation metrics are under constant debate and their true value in distinguishing better and worse systems is often called into question. We will come back to this discussion in Section 8.2.5; let us first get a better understanding of automatic evaluation methods.

8.2.1 Precision and Recall

How could we possibly expect a computer to assess the quality of a translation? All automatic evaluation metrics use the same trick. Each system translation is compared against one or more human translations of the same sentence. The human translations are called **reference translations**. We have argued at the beginning of this chapter that it is too much to ask of a machine translation system, or a human translator for that matter, to match a reference translation exactly. But a translation that is very similar to a reference translation is more likely to be

reference translation

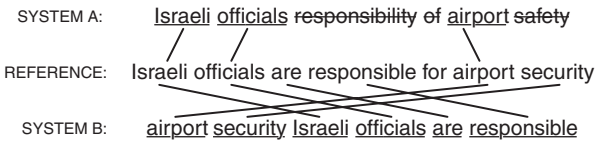

Figure 8.4 Matching words in the system output to the reference translation: System A has three correct words, while all six of System B's words are correct.

correct than one that differs substantially. The challenge for automatic evaluation metrics is to come up with a good similarity measure.

Let us start with an evaluation metric based on word matches. Check the first example in Figure 8.4. System A's output is *Israeli officials responsibility of airport safety*, which shares three words (*Israeli, officials* and *airport*) with the reference translation *Israeli officials are responsible for airport security*.

System A's output has six words in total, so three correct words out of six is a ratio of 50%. This type of metric is called **precision**. System B's output is *airport security Israeli officials are responsible*. The output has six words, and all them are also in the reference translation. Six correct out of six gives a nice 100% precision.

precision

There are clearly problems with System B's output, despite the perfect precision. First, the words are out of order, the phrase *airport security* should be moved to the end of the sentence. Focusing on word matches alone and ignoring their order has obvious short-comings. But also in terms of word matches, the translation is not perfect. The reference translation has one more word, *for*, so should we not consider this as well?

So, instead of computing how many of the words that a system generates are correct, we may compute how many of the words that a system *should* generate are correct. This metric is called **recall**. In contrast to precision, we divide the number of correct words by the length of the reference translation, instead of the length of the system output:

recall

$$\text{precision} = \frac{\text{correct}}{\text{output-length}} \tag{8.2}$$

$$\text{recall} = \frac{\text{correct}}{\text{reference-length}} \tag{8.3}$$

Both of these metrics can be deliberately tricked. We may produce translations only for words that we are sure of. The output will be very short, but we will have a very high precision (but low recall). Correspondingly, we may output all kinds of words, so the chance is high that we match all the words in the reference translation. The output will be very long, but we will have very high recall (but low precision).

Precision and recall are common metrics in natural language processing, and in some applications one is more important than the other.

Consider searching the web. The desired information may be contained in a large number of web pages. So, it is important that we get a few good results from a search engine and do not get confused by a large number of mismatches. Usually, there is no need to retrieve all pages. In this application, precision is more important than recall.

In machine translation, we are typically equally interested in precision and recall. We do not want to output wrong words, but we do not want to miss out on anything either. A common way to combine precision and recall is the **f-measure**, which is defined as the harmonic mean of the two metrics:

f-measure

$$\text{f-measure} = \frac{\text{precision} \times \text{recall}}{(\text{precision} + \text{recall})/2} \tag{8.4}$$

In our case, this can be reformulated as

$$\text{f-measure} = \frac{\text{correct}}{(\text{output-length} + \text{reference-length})/2} \tag{8.5}$$

Let us consider one more variation on this. **Position-independent error rate** (PER) is occasionally used in machine translation evaluation. It is similar to recall in that it uses the reference length as a divisor. It is an error rate, so we measure mismatches, not matches. To overcome the problem of too long translations, the metric also considers superfluous words that need to be deleted as wrong:

position-independent error rate

$$\text{PER} = 1 - \frac{\text{correct} - \max(0, \text{output-length} - \text{reference-length})}{\text{reference-length}} \tag{8.6}$$

In summary, the scores for the two systems using word-order sensitive precision, recall, and f-measure are:

Metric	System A	System B
precision	50%	57%
recall	43%	57%
f-measure	46%	57%
PER	57%	14%

8.2.2 Word Error Rate

Word error rate (WER), one of the first automatic evaluation metrics applied to statistical machine translation, is borrowed from speech recognition and takes word order into account. It employs the **Levenshtein distance**, which is defined as the minimum number of editing steps – insertions, deletions, and substitutions – needed to match two sequences.

word error rate

Levenshtein distance

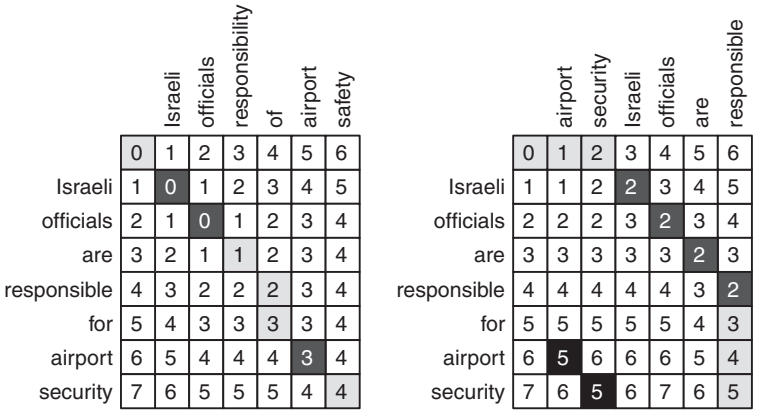

Figure 8.5 Two examples for the computation of the Levenshtein distance between output (on top) and reference translation (on the left): The distance is defined as the lowest-cost path (in grey) from the beginning of both sentences (top-left corner) to the end of both sentences (bottom-right corner), using word matches (cost 0, dark colors), or the editing steps of substitutions, insertions, and deletions (cost 1, light colors).

See Figure 8.5 for an illustration. The task of finding the minimum number of editing steps can be seen as finding the optimal path through the word alignment matrix of output sentence (across) and reference translation (down). Using a dynamic programming approach, we start at the top left corner (beginning of both sentences), and fill each point in the matrix with the cheapest cost of either:

match: if the point is a word match, take the cost from the point which is diagonally to the left-top;

substitution: if the point is not a word match, take the cost from the point which is diagonally to the left-top, add one;

insertion: take the cost from the point to the left, add one;

deletion: take the cost from the point above, add one.

Given the Levenshtein distance, we can compute the word error rate. Word error rate normalizes the number of editing steps by the length of the reference translation:

$$\text{WER} = \frac{\text{substitutions} + \text{insertions} + \text{deletions}}{\text{reference-length}} \tag{8.7}$$

The requirement of matching words in order may seem too harsh. Note that one of the human translations in Figure 8.1 was *This airport's security is the responsibility of the Israeli security officials*, which is a perfectly fine translation, but in the opposite order to the reference translation, so it will be marked with a very high word error rate.

In summary, the scores for the two systems using word error rate are:

Metric	System A	System B
word error rate (WER)	57%	71%

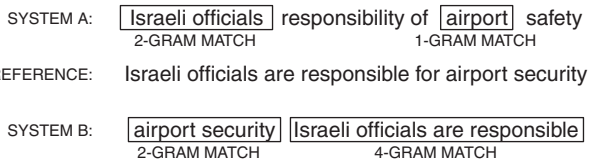

Figure 8.6 The BLEU score is based on n-gram matches with the reference translation.

8.2.3 BLEU: A Bilingual Evaluation Understudy

BLEU metric

The currently most popular automatic evaluation metric, the **BLEU metric**, has an elegant solution to the role of word order. It works similarly to position-independent word error rate, but considers matches of larger n-grams with the reference translation.

See Figure 8.6 for an illustration of n-gram matches in our previous example. System A's output matches are a 2-gram match for *Israeli officials* and a 1-gram match for *airport*. All output words of System B match: *airport security* is a 2-gram match and *Israeli officials are responsible* is a 4-gram match.

Given the n-gram matches, we can compute n-gram precision, i.e., the ratio of correct n-grams of a certain order n in relation to the total number of generated n-grams of that order:

- System A: 1-gram precision 3/6, 2-gram precision 1/5, 3-gram precision 0/4, 4-gram precision 0/3.
- System B: 1-gram precision 6/6, 2-gram precision 4/5, 3-gram precision 2/4, 4-gram precision 1/3.

The BLEU metric is defined as

$$\begin{aligned} \text{BLEU-n} &= \text{brevity-penalty } \exp \sum_{i=1}^{n} \lambda_i \log \text{precision}_i \\ \text{brevity-penalty} &= \min\left(1, \frac{\text{output-length}}{\text{reference-length}}\right) \end{aligned} \tag{8.8}$$

brevity penalty

The problem with precision-based metrics – no penalty for dropping words – is addressed by BLEU with a **brevity penalty**. The penalty reduces the score if the output is too short. The maximum order n for n-grams to be matched is typically set to 4. This metric is then called BLEU-4. Moreover, the weights λ_i for the different precisions are typically set to 1, which simplifies the BLEU-4 formula to

$$\text{BLEU-4} = \min\left(1, \frac{\text{output-length}}{\text{reference-length}}\right) \prod_{i=1}^{4} \text{precision}_i \tag{8.9}$$

Note that the BLEU score is 0 if any of the n-gram precisions is 0, meaning no n-grams of a particular length are matched anywhere in the output. Since n-gram precisions of 0 especially for 4-grams often occur

on the sentence level, BLEU scores are commonly computed over the entire test set.

Another innovation of the BLEU score is the use of **multiple reference translations**. Given the variability in translation, it is harsh to require matches of the system output against a single human reference translation. If multiple human reference translations are used, it is more likely that all acceptable translations of ambiguous parts of the sentences show up.

multiple reference translations

See Figure 8.7 for an example. Originally, System A did not get any credit for the output of *responsibility of*, which is not a wrong translation. Indeed, the words do show up in one of the reference translations.

The use of multiple reference translations works as follows. If an n-gram in the output has a match in any of the reference translations, it is counted as correct. If an n-gram occurs multiple times in the output (for instance the English world *the* often shows up repeatedly), it has to occur in a single reference translation the same number of times for all occurrences to be marked as correct. If reference translations have fewer occurrences of the n-gram, it is marked as correct only that many times.

Multiple reference translations complicate the issue of reference length. In multiple-reference BLEU for each output sentence the closest length of any of the reference translations is determined and taken as the reference length. If two reference lengths are equally close, but one is shorter and one longer, the shorter one is taken. For instance, given an output length of 10 and lengths of reference sentences 8, 9, 11, and 15, the reference length for that sentence is 9 (both 9 and 11 are equally close, but 9 is smaller).

In summary, the various precisions, brevity penalty, and BLEU scores for two example sentences from Figure 8.6 on the facing page are (see Figure 8.7 for the multiple references):

	Single reference		Multiple reference	
Metric	System A	System B	System A	System B
precision (1-gram)	3/6	6/6	5/6	6/6
precision (2-gram)	1/5	4/5	2/5	4/5
precision (3-gram)	0/4	2/4	0/4	2/4
precision (4-gram)	0/3	1/3	0/3	1/3
brevity penalty	6/7	6/7	6/7	6/7
BLEU-1	42%	86%	71%	86%
BLEU-2	9%	69%	29%	69%
BLEU-3	0%	34%	0%	34%
BLEU-4	0%	11%	0%	11%

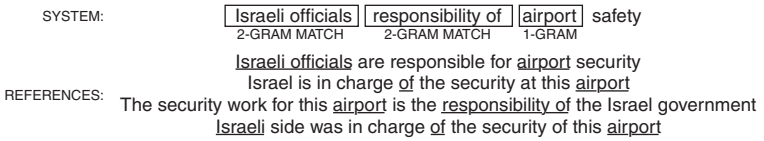

Figure 8.7 Additional n-gram matches by using multiple reference translations, accounting for variability in acceptable translations.

8.2.4 METEOR

Recently, many variations and extensions to the BLEU metric have been proposed. One point of discussion, for instance, is the role of precision and recall in machine translation evaluation. The case is made that recall is more important to ensure that the complete meaning is captured by the output.

METEOR metric

One more recent metric, **METEOR**, incorporates a stronger emphasis on recall, and also introduces a couple of novel ideas. One perceived flaw of BLEU is that it gives no credit to near matches. Recall that one of our example system outputs used the noun *responsibility*, but the reference used the adjective *responsible*. Both carry the same meaning, but since the words are not the same, BLEU counts this as an error. By stemming the two words, i.e., reducing them to their stems, we are able to match them.

Another way to detect near misses is using synonyms, or semantically closely related words. The human translations in the lead example of this chapter (Figure 8.1) give some examples for this. Translators varied in their use of *security* and *safety*, as well as *responsibility* and *charge*. Often, these word choices are irrelevant in bringing across the meaning of the sentences and should not be penalized.

METEOR incorporates the use of stemming and synonyms by first matching the surface forms of the words, and then backing off to stems and finally semantic classes. The latter are determined using Wordnet, a popular ontology of English words that also exists for other languages.

The main drawback of METEOR is that its method and formula for computing a score is much more complicated that BLEU's. The matching process involves computationally expensive word alignment. There are many more parameters – such as the relative weight of recall to precision, the weight for stemming or synonym matches – that have to be tuned.

8.2.5 The Evaluation Debate

The use of automatic evaluation metrics in machine translation is under constant debate in the research community. It seems to be hard to believe that simplistic metrics such as the BLEU score do properly reflect differences in meaning between system output and reference

translations (or input, for that matter). Another reason for this debate is that these automatic metrics are almost exclusively used by *statistical* machine translation researchers, and their claims are often questioned by practitioners of machine translation by different means.

The main points of **critique** are: critique of BLEU

- BLEU ignores the relative relevance of different words. Some words matter more than others. One of the most glaring examples is the word *not*, which, if omitted, will cause very misleading translations. Names and core concepts are also important words, much more so than, e.g., determiners and punctuation, which are often irrelevant. However, all words are treated the same way by the metrics we presented here.
- BLEU operates only on a very local level and does not address overall grammatical coherence. System output may look good on an n-gram basis, but very muddled beyond that. There is a suspicion that this biases the metric in favor of phrase-based statistical systems, which are good at producing good n-grams, but less able to produce grammatically coherent sentences.
- The actual BLEU scores are meaningless. Nobody knows what a BLEU score of 30% means, since the actual number depends on many factors, such as the number of reference translations, the language pair, the domain, and even the tokenization scheme used to break up the output and reference into words.
- Recent experiments computed so-called human BLEU scores, where a human reference translation was scored against other human reference translations. Such human BLEU scores are barely higher (if at all) than BLEU scores computed for machine translation output, even though the human translations are of much higher quality.

Many of these arguments were also brought up initially by current users of BLEU, and all of them also apply broadly to any other automatic evaluation metric. There are counter-arguments; the most convincing is shown in Figure 8.8. The graph in the figure plots system performance as measured by automatic scores against human judgement scores for submissions to the 2002 machine translation evaluation on Arabic–English, organized by NIST.

In this analysis, systems with low automatic scores also received low human judgement scores, and systems with high automatic scores also received high human judgement scores. What this shows is a high **correlation** between human and automatic scores. This is what we expect from a good automatic evaluation metric, and this is what evaluation metrics should be assessed on. human–automatic correlation

8.2.6 Evaluation of Evaluation Metrics

The most widely used method for computing the correlation between two metrics is the **Pearson correlation coefficient**. Formally, we are Pearson correlation coefficient

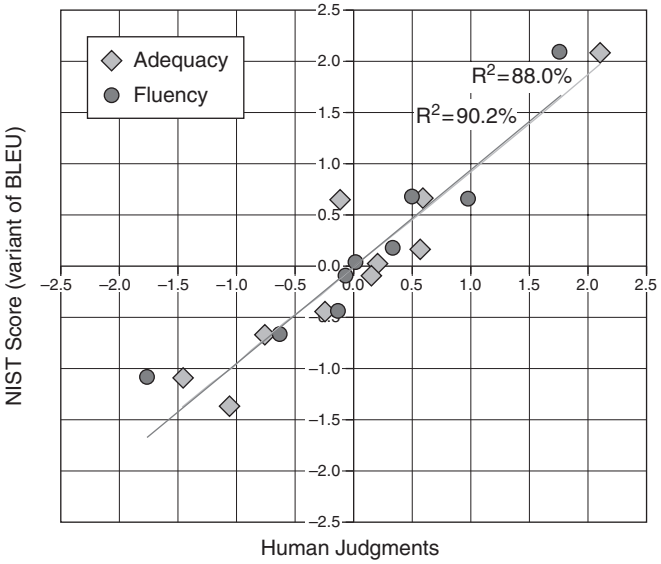

Figure 8.8 Correlation between an automatic metric (here: NIST score) and human judgment (fluency, adequacy). Illustration by George Doddington.

Correlation	Negative	Positive
small	−0.29 to −0.10	0.10 to 0.29
medium	−0.49 to −0.30	0.30 to 0.49
large	−1.00 to −0.50	0.50 to 1.00

Figure 8.9 Typical interpretation of the correlation coefficient r_{xy}.

faced with a set of data points $\{(x_i, y_i)\}$ that contain values for two variables x, y. The Pearson correlation coefficient r_{xy} between the two variables is defined as

$$r_{xy} = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{(n-1)\, s_x\, s_y} \tag{8.10}$$

To compute the coefficient r_{xy} we first need to compute the sample means $\bar{x}$, $\bar{y}$ and the sample variances s_x, s_y of the two variables x and y:

$$\begin{aligned} \bar{x} &= \frac{1}{n} \sum_{i=1}^{n} x_i \\ s_x^2 &= \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^2 \end{aligned} \tag{8.11}$$

What counts as a good correlation between perfect correlation ($r_{xy} = 1$) and total independence between the variables ($r_{xy} = 0$) is anybody's guess. Figure 8.9 gives a typical interpretation of the correlation coefficient. If our goal is to compare different automatic evaluation metrics with respect to their correlation with a manual metric, the answer is straightforward: a higher correlation is better.

8.2.7 Evidence of Shortcomings of Automatic Metrics

shortcomings of BLEU

Recent evaluation campaigns have revealed exceptions to the correlation of manual and automatic scores for some special cases.

In the 2005 NIST evaluations on Arabic–English, one of the submissions was a system that was improved by human post-editing. The post-editing was done by monolingual speakers of the target language with no knowledge of the source language. The post-editing effort led to only small increases in the automatic BLEU score, but to large improvements in both fluency and adequacy judgments in the manual evaluation. See Figure 8.10a,b.

Secondly, in an experiment comparing a commercial rule-based machine translation system with two instances of a statistical system (one trained on the full data set, the other on 1/64th of it), the automatic BLEU score failed to reflect the human judgment. The worse statistical system, although given much lower judgments by humans, still achieved a higher BLEU score than the rule-based system. Human evaluators scored the rule-based system and the better statistical system similarly; the BLEU scores, however, were 18% and 30%, respectively. See Figure 8.10c.

The WMT 2006 evaluation campaign confirmed the latter result – again a rule-based system participated along with a range of statistical systems. Interestingly, the correlation of scores was much higher when the systems were tested on out-of-domain test data. While the rule-based system was not developed for a specific domain, the statistical

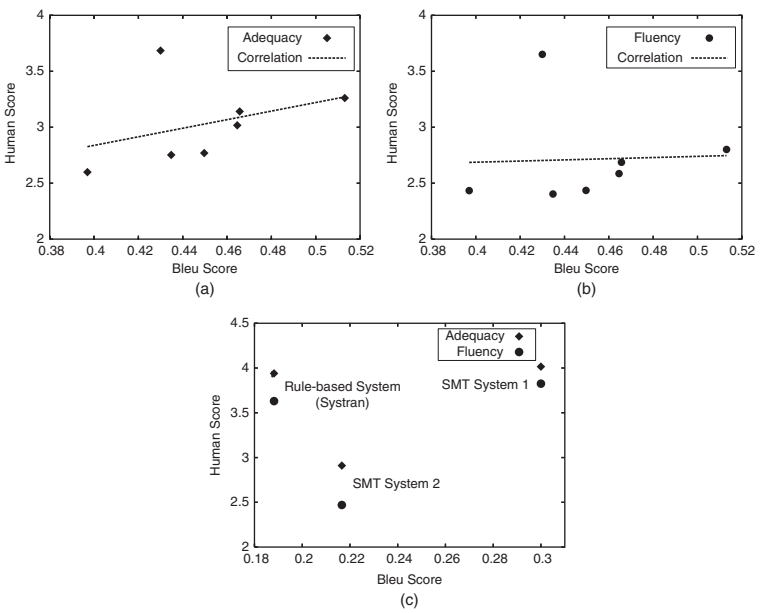

Figure 8.10 Examples of lack of correlation between BLEU and manual judgment. Manually post-edited machine translation is scored low by BLEU, but high by humans, both in terms of adequacy (a) and fluency (b). A rule-based system receives a much lower BLEU score than a comparable statistical system that is trained and evaluated on the Europarl corpus (c).

systems were trained on Europarl data, but the out-of-domain test data set was political and economic commentary. This finding seems to suggest that part of the explanation for the lack of correlation is that automatic scores are overly literal and reward the right choice of jargon much more strongly than human evaluators.

The state of the current debate on automatic evaluation metrics evolves around a general consensus that automatic metrics are an essential tool for system development of statistical machine translation systems, but not fully suited to computing scores that allow us to rank systems of different types against each other. Developing evaluation metrics for this purpose is still an open challenge to the research community.

8.3 Hypothesis Testing

Machine translation evaluation is typically carried out in the following manner. Two (or more) machine translation systems produce translations for a set of test sentences. For each system's output an evaluation score is obtained, either by eliciting human judgments, or by computing automatic scores using reference translations.

If this leads to different scores for different systems, we would like to conclude that one system is better than the other. But there may also be another explanation for the different scores. Both systems are actually performing equally well, and the two different scores are just a result of random variation in this particular test set. This explanation is called the **null hypothesis**, as opposed to the hypothesis that one system is in fact better than the other.

null hypothesis

hypothesis testing

The task of **hypothesis testing** is to decide which of the two explanations is more likely to be true. Note that random variation may account for any difference in scores, but it is a less probable explanation for large differences. Hypothesis testing will never give us certainty that observed score differences are true differences, but it allows us to set arbitrary significance levels.

If there is less than 1% chance that the difference in score between two systems is due to random variation of two equally well performing systems, we say that they are different with 99% **statistical significance**. Typically, researchers require 95% or 99% statistical significance. This is often also expressed as a **p-level**, the probability of an erroneous conclusion. A p-level of $p < 0.01$ is another way to express 99% statistical significance.

statistical significance

p-level

In the case of evaluating a single system, we are interested in the following question: Given the measured evaluation score x, what is the true evaluation score? Or, to phrase it as a question that we are actually

able to answer: With a statistical significance of, say, 95%, what is the range of scores that includes the true score? This range is called the **confidence interval**. It is typically computed with a distance d around the measured score x, i.e., the interval $[x-d, x+d]$.

confidence interval

8.3.1 Computing Confidence Intervals

Let us consider the simplest form of evaluation that we have presented so far. Given the system output, a human evaluator judges each sentence translation to be either correct or false. If, say, 100 sentence translations are evaluated, and 30 are found correct, what can we say about the true translation score of the system? Our best guess is 30%, but that may be a few percent off. How much off, is the question to be answered by statistical significance tests.

Given a set of n sentences, we can compute the sample mean $\bar{x}$ and variance s^2 of the individual sentence scores x_i:

$$\begin{aligned} \bar{x} &= \frac{1}{n}\sum_{i=1}^{n} x_i \\ s^2 &= \frac{1}{n-1}\sum_{i=1}^{n}(x_i-\bar{x})^2 \end{aligned} \tag{8.12}$$

What we are really interested in is, however, the true mean μ (the true translation score). Let us assume that the sentence scores are distributed according to the normal distribution. See Figure 8.11 for an illustration. Given the sample mean $\bar{x}$ and sample variance s^2, we estimate the probability distribution for true translation quality. We are now interested in a confidence interval $[\bar{x}-d, \bar{x}+d]$ around the mean sentence score. The true translation quality (or the true mean μ) lies within the confidence interval with a certain probability (the statistical significance level).

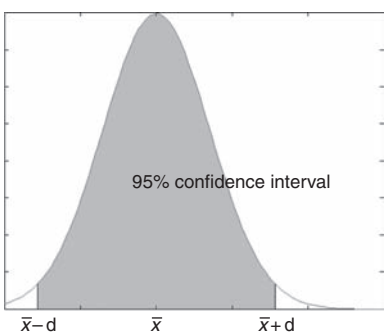

Figure 8.11 With probability $q = 0.95$, the true score lies in an interval $[\bar{x}-d, \bar{x}+d]$ around the sample score $\bar{x}$.

Note the relationship between the degree of statistical significance and the confidence interval. The degree of statistical significance is indicated by the fraction of the area under the curve that is shaded. The confidence interval is indicated by the boundaries on the x-axis. A larger statistical significance, a large shaded area, pushes the boundaries of the confidence interval outward.

Since we do not know the true mean μ and variance σ^2, we cannot model the distribution of sentence scores with the normal distribution. However, we can use Student's t-distribution, which approximates the normal distribution for large n.

The function that maps between a confidence interval $[\bar{x} - d, \bar{x} + d]$ and the statistical significance level can be obtained by integrating over the distribution. However, in case of Student's t-distribution, the integral cannot be evaluated in closed form, but we can use numerical methods.

The size of the confidence interval can be computed by

$$d = t \cdot \frac{s}{\sqrt{n}} \tag{8.13}$$

The factor t depends on the desired p-level of statistical significance and the sample size. See Figure 8.12 for typical values.

Let us review the assumptions underlying this method of defining the confidence intervals. We randomly draw test sentences and score them with the system. The confidence interval gives an indication of what the likely true average error is, if it were computed on an infinite number of test sentences.

8.3.2 Pairwise Comparison

We are usually not interested in the absolute value of evaluation scores. As we have argued, these scores are often hard to interpret. More typically, we are interested in the comparison of two (or more) systems. We want to render a judgment, to decide whether one system is indeed better than the other. This may involve comparing different systems in a competition or, as is more often the case, comparing a baseline system against a change that we want to assess.

We may use the confidence intervals that we computed in the previous section to assess the statistical significance of the difference in

	Test sample size			
Significance level	100	300	600	∞
99%	2.6259	2.5923	2.5841	2.5759
95%	1.9849	1.9679	1.9639	1.9600
90%	1.6602	1.6499	1.6474	1.6449

Figure 8.12 Values for t for different sizes and significance levels (to be used in Formula 8.13).

scores. If the confidence intervals do not overlap, we are able to state that the two systems do in fact have different performance at the given statistical significance level.

Alternatively, instead of using confidence intervals, we may want to compute the statistical significance of score differences directly for that **pairwise comparison**. For instance, given a test set of 100 sentences, if we find one system doing better on 40 sentences, and worse on 60 sentences, is that difference statistically significant? pairwise comparison

The **sign test** allows us to compute how likely it is that two equal systems would come up with such a sample of performance differences. The binomial distribution is used to model such a scenario. If system A is better at translating a sentence with probability p_A, and system B is better with probability p_B ($=1-p_A$), then the probability of system A being better on $k = 40$ sentences out of a sample of $n = 100$ sentences is sign test

$$\binom{n}{k} p_A^k \, p_B^{n-k} = \frac{n!}{k!\,(n-k)!} \, p_A^k \, p_B^{n-k} \tag{8.14}$$

According to the null hypothesis $p_A = p_B = 0.5$, so the formula simplifies to

$$\binom{n}{k} p^k \, (1-p)^{n-k} = \binom{n}{k} 0.5^n = \frac{n!}{k!\,(n-k)!} \, 0.5^n \tag{8.15}$$

The particular outcome of system A being better on exactly 40 sentences is unlikely, so we want to compute the whole range of $k \in [0, 40]$, when considering the null hypothesis:

$$p(0..k; n) = \sum_{i=0}^{k} \binom{n}{i} 0.5^n \tag{8.16}$$

This formula computes the probability that, given two equally well performing systems, one system is better on up to k out of n sentences. If such an outcome is unlikely (say, with a p-level of $p \leq 0.05$), we reject the null hypothesis, and conclude that the difference reflects a difference in quality between the systems. Note that when using this formula, we ignore sentences where the two systems are scored the same.

For our example, we plug the numbers $n = 100$ and $k = 40$ into the formula for the sign test and we get $p = 0.0569$. So, we cannot say with statistical significance at a p-level of $p \leq 0.05$ that the systems differ. See Figure 8.13 for more illustrations of this formula. It gives the minimum number of k out of n to achieve statistical significance at different p-levels. In our example of 100 sentences, one system has to be better in at least 61 cases (and the other better in at most 39 cases) to achieve statistical significance at p-level $p \leq 0.05$.

n	$p \leq 0.01$		$p \leq 0.05$		$p \leq 0.10$	
5	–	–	–	–	$k = 5$	$\frac{k}{n} = 1.00$
10	$k = 10$	$\frac{k}{n} = 1.00$	$k \geq 9$	$\frac{k}{n} \geq 0.90$	$k \geq 9$	$\frac{k}{n} \geq 0.90$
20	$k \geq 17$	$\frac{k}{n} \geq 0.85$	$k \geq 15$	$\frac{k}{n} \geq 0.75$	$k \geq 15$	$\frac{k}{n} \geq 0.75$
50	$k \geq 35$	$\frac{k}{n} \geq 0.70$	$k \geq 33$	$\frac{k}{n} \geq 0.66$	$k \geq 32$	$\frac{k}{n} \geq 0.64$
100	$k \geq 64$	$\frac{k}{n} \geq 0.64$	$k \geq 61$	$\frac{k}{n} \geq 0.61$	$k \geq 59$	$\frac{k}{n} \geq 0.59$

Figure 8.13 Examples for the sign test: Given a number of sentences *n*, one system has to be better in at least *k* sentences to achieve statistical significance at the specified p-level.

8.3.3 Bootstrap Resampling

We have described methods for computing confidence intervals and statistical significance of test score differences. However, the methods operate under the assumption that we can compute scores for single sentences. This assumption does not hold for the canonical metric in machine translation, the BLEU score, since it is not computed on the sentence level.

One solution to this problem is to break up the test set into blocks of, say, 10–20 sentences, and compute BLEU scores for each. Then we are able to resort to the presented methods, and for instance apply the sign test on these blocks. If one system outperforms the other in translating significantly more blocks better, the difference is statistically significant.

bootstrap resampling

An alternative method is called **bootstrap resampling**. Let us first consider the case of estimating confidence intervals. The pairwise comparison task is very similar.

Consider the following: Let us say that we are using a test set of 2,000 sentences that we sampled from a large collection of available test sentences. We then proceed to compute the BLEU score for this set. But how likely is that score to be representative of the true BLEU score computed on a test of near-infinite size?

If we were to repeatedly sample test sets of 2,000 sentences, and compute the BLEU score on each of them, we would get a distribution of scores that looks like the bell curve in Figure 8.11. With enough test set samples, say 1,000, we can then empirically determine the 95% confidence interval. By ignoring the 25 highest and 25 lowest BLEU scores, we are left with an interval that contains 95% of the scores.

To restate the argument: if we pick one of the test set samples at random, then with 95% probability it will be in the interval between the extreme tails of the distribution. Hence, a truly representative test set sample will also lie with 95% probability in this interval.

Of course, taking 1,000 test sets of 2,000 sentences means translating 2 million sentences, and if we were able to do that, we could most likely come up with a tighter confidence interval using the sign test on blocks, as mentioned above.

Therefore, we apply the following trick. We sample the 1,000 test sets from the *same* 2,000 sentences of the initial test set, with replacement. Since we are allowed to take the same sentences more than once, we will likely come up with 1,000 different test sets, and therefore 1,000 different test scores. We then move on and compute the confidence interval, as if these sets were truly independent samples.

We will not go into the theoretical arguments about bootstrap resampling here; the reader is referred to Efron and Tibshirani [1993]. Intuitively, bootstrap resampling uses the variability in the test sets, as does the sign test, to draw conclusions about statistical significance. If the system translated most parts of the test set with very similar performance, then we are more likely to trust the resulting score, as opposed to a test set translation with parts of widely different performance.

The application of bootstrap resampling to the pairwise comparison of different systems is straightforward. We compute both systems' scores on the resampled test set, and check which system is better. If one system is better on at least 950 samples, then it is deemed to be statistically significantly better at the $p \leq 0.05$ p-level.

8.4 Task-Oriented Evaluation

So far we have discussed various manual and automatic evaluation metrics that aim to judge the quality – or relative quality in the form of a ranking – of machine translation output. These metrics take into account that most sentences will have some errors in them, and they give some measurement of the error rate. The reduction of this error rate is the goal of research activity in the field of machine translation.

But maybe the question *How good is machine translation?* is the wrong question to ask, maybe a better question is *Is machine translation good enough?* Machine translation is not an end itself; it is used to support some kind of task, perhaps supporting the efforts of a human translator to more efficiently translate documents for publication, or perhaps helping someone to understand the contents of a document in an unknown foreign language.

If these are the uses of machine translation, then machine translation will ultimately be evaluated in the marketplace on how well it supports these tasks.

8.4.1 Cost of Post-Editing

Consider this scenario of the use of machine translation systems: first a foreign document is translated with a machine translation system, then a human translator corrects the errors, and submits a final high-quality translation for publication.

For this scenario to work, the human translator will only make a few corrections. An evaluation metric that reflects this may count the minimum number of required corrections.

Levenshtein distance

We have already presented one metric that compares the system output with a reference translation in terms of editing steps. **Levenshtein distance** (Section 8.2.2 on page 224) counted the number of insertions, deletions, and substitutions needed to transform the system output to match the reference translation.

One disadvantage of the Levenshtein distance is that mismatches in word order require the deletion and re-insertion of the misplaced words. We may remedy this by adding an editing step that allows the movement of word sequences from one part of the output to another. This is something a human post-editor would do with the cut-and-paste function of a word processor.

translation error rate
cover disjoint error rate

Some evaluation metrics based on such editing steps have been proposed, including **translation error rate** (TER) and **cover disjoint error rate** (CDER). The metrics are based on the Levenshtein distance, but add block movement (also called a jump) as an editing step. The computation of TER scores is actually not trivial: finding the shortest sequence of editing steps is a computationally hard problem. It is, in fact, NP-complete. CDER is computationally less complex, due to a relaxation of the alignment requirements.

The metrics TER and CDER still suffer from one short-coming. The system output may be an acceptable translation, but different from the reference translation. Hence, the error measured by these metrics does not do justice to the system.

human translation error rate

Alternatively, we may ask a human evaluator to post-edit the system output and count how many editing steps were undertaken. One example of a metric that is designed in such a manner is the **human translation error rate** (HTER), used in recent DARPA evaluations. Here, a human annotator has to find the minimum number of insertions, deletions, substitutions, and shifts to convert the system output into an acceptable translation. HTER is defined as the number of editing steps divided by the number of words in the acceptable translation.

Finding the minimum number of editing steps is often a time-consuming task. It does not reflect what a human post-editor would do. A post-editor would try to spend as little time as possible to correct the sentence. This may involve more editing steps than indicated by HTER scores. If the output is very muddled, human translators are justified in simply deleting the machine translation and writing a new translation from scratch, instead of fiddling with the broken output.

post-editing time

When considering **post-editing time**, one also has to take into consideration the time spent on reading the system output. If the human

translator reads buggy machine translation output, discards it, and then writes his own translation, he will have spent more time than if he were not using machine translation technology at all.

Error-prone machine output may not only result in more time spent on translation, it may also disrupt the work flow of translators. This added frustration may be one reason today's human translators rarely use full-fledged machine translation systems to support their work.

Note that a good translation tool for human translators is sometimes better off not providing any translation at all. See also our discussion of applications of machine translation technology to aid translators in Section 1.3.

8.4.2 Content Understanding Tests

Let us now consider metrics that try to assess whether machine translation output is good enough that a human reader is able to **understand the content** of the translated document.

understanding tests

Note that the bar for machine translation is lower here. A translation that has muddled grammar and misplaced or missing function words may still be fully informative. We are constantly confronted with incorrect English – most spontaneous speech is ungrammatical – but that does not prevent us from understanding it.

Extracting meaning from text requires increasing levels of understanding:

1. Basic facts: Who is talked about? Where and when do the events take place? Detect names, numbers, and dates. What is the main gist of the story?
2. Detailed relationships between the elements of the text: How do the entities relate to each other? What is the order of events? What causality exists between them?
3. Nuance and author intent: How are the facts characterized and emphasized? Why did the author express them in the given manner? What is the subtext?

If we hand a translated text to human evaluators and ask them questions that range in difficulty, we can measure how well they are able to understand the text. We then take the ratio of correctly answered questions as a judgment of how well the machine translation system retained the meaning of the text. The question may be broken down by the levels mentioned above or other categories. We may also measure the time spent on answering the questions – good machine translation output should be faster to read and comprehend.

Of course, this test measures not only the capability of the machine translation system to provide understandable text, but also the capability

of the human evaluator. This cuts both ways. The evaluator's reading proficiency and ability to answer test questions in his native language limits his ability to answer such questions on machine translated output. On the other hand, a clever test subject with broad knowledge about the subject matter at hand may be able to fill in missing or mistranslated content.

8.5 Summary

8.5.1 Core Concepts

This chapter addressed the issue of **evaluating** machine translation performance. This is a hard problem, since a foreign sentence may have many different correct translations. We may be able to provide some **reference translations**, but we cannot expect machine translation systems (or even human translators) to match these exactly.

Manual evaluation metrics ask a human evaluator to render an assessment of the translation performance, say, on a sentence-by-sentence basis. **Adequacy** measures how much meaning is retained in the translation, while **fluency** measures whether the output is good English. Different human evaluators tend to differ in their leniency or harshness when assigning scores, so we have to **normalize** these to make them comparable. Human evaluators are more consistent in **ranking** different translated sentences against each other than in assigning absolute scores.

We would like to use evaluation metrics that are **low cost** in terms of time and money spent, result in **meaningful** numbers, and are **consistent**. By consistent we mean **inter-annotator agreement** – different evaluators should come to the same assessment – and **stable** with respect to different parts of the text. Most importantly, we expect a metric to be **correct**, i.e., indicating actual performance and ranking better systems higher than worse ones. Besides quality metrics, we may also consider, in the evaluation of machine translation systems, matters such as **speed** and **size** of the system, ease of **integration** into an application environment, and support for **domain adaptation** and **customization**.

Since manual evaluation is very labor intensive, it would be preferable to have reliable automatic evaluation metrics for daily use. When comparing system output with reference translations, we may consider the **precision** and **recall** of words. **Word error rate** (WER) uses the **Levenshtein distance** to compute the minimum number of edits needed to get from the system output to the reference translation.

While WER leads to very low scores when the word order is wrong, **position-independent word error rate** (PER) ignores word order when matching output and reference. The currently most popular metric is **BLEU**, which uses n-gram matches with the reference and also makes use of **multiple reference translations** to account for variability. The **METEOR** metric also considers matches on the lemma-level and synonym matches. Evaluation of evaluation metrics is done by measuring correlation with human judgments, which may be computed using the **Pearson correlation coefficient**.

With **hypothesis testing**, we are able to assess the **statistical significance** of score differences. We distinguish genuine performance differences from the **null hypothesis**, i.e., random variation in evaluation scores of systems performing equally well. We assess significance with a minimum specified **p-level** of certainty. We may compute a **confidence interval** around the measured score, but more often we are interested in the **pairwise comparison** of systems. The **sign test** is one method that is used for such sentence-level scores. It does not easily apply to the BLEU score, for which we may resort to **bootstrap resampling**.

Task-oriented evaluation metrics attempt to directly measure the utility of machine translation for performing a specific task. If the task is to produce high-quality translations, we may measure the **post-editing time** required to correct machine translation output. In a similar vein, the **translation error rate** measures the number of editing steps required to reach a reference translation, and the **human translation error rate** measures how many editing steps a human post-editor has to perform to reach an acceptable translation. Extracting information from translated text material is the motivation for **understanding tests**, which measure how well a human evaluator is able to answer questions about a foreign text given machine translation output.

8.5.2 Further Reading

Evaluation campaigns – The first machine translation evaluation campaign, in which both statistical and traditional rule-based machine translation systems participated, was organized by ARPA in the early 1990s. White *et al.* [1994] present results and discuss in detail the experience with different evaluation strategies. The straightforward application of metrics for human translators proved difficult, and was abandoned along with measurements of productivity of human-assisted translation, which hinges to a large degree on the quality of the support

tools and the level of expertise of the translator. Ultimately, only reading comprehension tests with multiple choice questions and adequacy and fluency judgments were used in the final evaluation. Hamon *et al.* [2007b] discuss the evaluation of a speech-to-speech translation system.

Manual metrics – King *et al.* [2003] present a large range of evaluation metrics for machine translation systems that go well beyond the translation quality measures to which we devoted the bulk of this chapter. Miller and Vanni [2005] propose *clarity* and *coherence* as manual metrics. Reeder [2004] shows the correlation between fluency and the number of words it takes to distinguish between human and machine translations. Grading standards for essays from foreign language learners may be used for machine translation evaluation. Using these standards reveals that machine translation has trouble with basic levels, but scores relatively high in advanced categories [Reeder, 2006a]. A manual metric that can be automated is one that asks for specific translation errors – the questions may be based on past errors [Uchimoto *et al.*, 2007]. Vilar *et al.* [2007a] argue for pairwise system comparisons as a metric, which leads to higher inter- and intra-annotator agreement [Callison-Burch *et al.*, 2007].

Task-based metrics – Task-based evaluation of machine translation tests the usefulness of machine translation directly, for instance the ability of human consumers of machine translation output to answer who, when, and where questions [Voss and Tate, 2006]. Jones *et al.* [2006] use a foreign language learner test for measuring speech-to-speech machine translation performance. The adaptation of such a test (the Defense Language Proficiency Test, DLPT) for machine translation revealed that Arabic–English machine translation performance achieves a passing grade up to level 2+, but performs relatively weakly on basic levels.

Word error rate – Word error rate was first used for the evaluation of statistical machine translation by Tillmann *et al.* [1997], who also introduce the position-independent error rate. Allowing block movement [Leusch *et al.*, 2003] leads to the definition of the CDER metric [Leusch *et al.*, 2006]. TER allows for arbitrary block movements [Snover *et al.*, 2006]. MAXSIM uses an efficient polynomial matching algorithm that also uses lemma and part-of-speech tag matches [Chan and Ng, 2008]. The way automatic evaluation metrics work also depends on the tokenization of system output and reference translations [Leusch *et al.*, 2005].

N-gram matching metrics – The BLEU evaluation metric is based on n-grams not words [Papineni *et al.*, 2001]. Several variants of n-gram matching have been proposed: weighting n-grams based on their frequency [Babych and Hartley, 2004], or other complexity metrics

[Babych *et al.*, 2004]. GTM is based on precision and recall [Melamed *et al.*, 2003; Turian *et al.*, 2003]. Echizen-ya and Araki [2007] propose IMPACT, which is more sensitive to the longest matching n-grams. A metric may benefit from using an explicit alignment of system output and reference while maintaining the advantages of n-gram based methods such as BLEU [Liu and Gildea, 2006] and by training such a metric to correlate to human judgment [Liu and Gildea, 2007]. Lavie *et al.* [2004] emphasize the importance of recall and stemmed matches in evaluation, which led to the development of the METEOR metric [Banerjee and Lavie, 2005; Lavie and Agarwal, 2007]. Partial credit for stemmed matches may also be applied to BLEU and TER [Agarwal and Lavie, 2008].

Syntax-based metrics – Metrics may be sensitive to syntactic information [Liu and Gildea, 2005]. A more syntactic approach to machine translation may consider the matching of dependency structures of reference and system output [Owczarzak *et al.*, 2007a], possibly including dependency labels [Owczarzak *et al.*, 2007b]. A wide range of linguistic features for evaluation is explored by Giménez and Màrquez [2007b]. A method based on latent semantic analysis has been proposed for machine translation evaluation [Reeder, 2006b].

Analytical metrics – Automatic metrics may also help in pinpointing the type of errors that machine translation systems make. For instance the relationship of WER and PER indicates the severity of reordering problems, and the relationship of BLEU on stemmed words and surface forms indicates the need for better morphological generation [Popovic *et al.*, 2006]. We may also gain insight into the types of errors by examining word error rates for different parts of speech [Popovic and Ney, 2007].

Evaluation without reference – Some researchers investigate methods that allow automatic evaluation without the use of a reference translation. Gamon *et al.* [2005] train a model that combines the language model probability of the output with syntactic and semantic features. Albrecht and Hwa [2007b, 2008] propose to learn a metric from human judgments and pseudo-reference translations, which are generated by using other machine translation systems that are not necessarily better.

Correlation of automatic and manual metrics – The credibility of automatic evaluation metrics rests on their correlation with reliable human judgments. Coughlin [2003] finds evidence in support of BLEU in a total of 124 evaluations for many European language pairs. Other evaluation campaigns continuously assess correlation of human and automatic metrics, such as in the CESTA campaign [Surcin *et al.*, 2005; Hamon *et al.*, 2007a]. Yasuda *et al.* [2003] and Finch *et al.* [2004]

investigate the required number of reference translations. The number of reference translations may be increased by paraphrasing [Finch *et al.*, 2004; Owczarzak *et al.*, 2006a]. The same idea is behind changing the reference translation by paraphrasing to make it more similar to the reference [Kauchak and Barzilay, 2006], or to attempt to paraphrase unmatched words in the system output [Zhou *et al.*, 2006]. Hamon and Mostefa [2008] find that the quality of the reference translation is not very important. Akiba *et al.* [2003] show strong correlation for BLEU only if the systems are of similar type. BLEU tends to correlate less when comparing human translators with machine translation systems [Culy and Riehemann, 2003; Popescu-Belis, 2003], or when comparing statistical and rule-based systems [Callison-Burch *et al.*, 2006b]. Amigó *et al.* [2006] find that the relationship between manual metrics that measure human acceptability and the automatic metrics that check the similarity of system output with human translations is a bit more complex.

Trained metrics – Albrecht and Hwa [2007a] argue for the general advantages of learning evaluation metrics from a large number of features, although Sun *et al.* [2008] point out that carefully designed features may be more important. Jones and Rusk [2000] propose a method that learns automatically to distinguish human translations from machine translations. Since in practice the purpose of evaluation is to distinguish good translations from bad translations, it may be beneficial to view evaluation as a ranking task [Ye *et al.*, 2007b; Duh, 2008]. Lin and Och [2004] propose a metric for the evaluation of evaluation metrics, which does not require human judgment data for correlation. The metric is based on the rank given to a reference translation among machine translations. Multiple metrics may be combined uniformly [Giménez and Màrquez, 2008b], or by adding metrics greedily until no improvement is seen [Giménez and Màrquez, 2008a].

Difficulty to translate – The difficulty of translating a text may depend on many factors, such as source, genre, or dialect, some of which may be determined automatically [Kirchhoff *et al.*, 2007]. Uchimoto *et al.* [2005] suggests using back-translation to assess which input words will cause problems, and then prompting the user of an interactive machine translation system to rephrase the input.

Statistical significance – Our description of the bootstrap resampling method [Efron and Tibshirani, 1993] for estimating statistical significance follows the description by Koehn [2004b]. For further comments on this technique, see the work by Riezler and Maxwell [2005]. Estrella *et al.* [2007] examine how big the test set needs to be for a reliable comparison of different machine translation systems.

8.5.3 Exercises

1. (⋆) Consider the following system output and reference translation:
   Reference: *The large dog chased the man across the street.*
   System: *The big dog chases a man across the street.*
   (a) Draw a matrix with system words on one axis and reference words on the other (as in Figure 8.5). Compute the word error rate.
   (b) Determine the precision for unigrams to 4-grams, and compute the BLEU score (ignoring the brevity penalty).
   (c) One suggested change to n-gram metrics is the use of partial credit for synonym matches or for the right lemma but wrong morphological form. Compute an adapted BLEU score where synonym and lemma matches count as 50% correct. State your assumptions about what constitutes a synonym.
2. (⋆) We want to test how many sentences a machine translation system translates correctly. We use test sets of various sizes and count how many sentences are correct. Compute the 90%, 95% and 99% confidence intervals for the following cases:

Size of test set	Correct
100 sentences	77 sentences
300 sentences	231 sentences
1000 sentences	765 sentences

3. (⋆⋆) Implement a bootstrap resampling method to measure the confidence intervals for the test sets in Question 2. How would you empirically test if bootstrap resampling is reliable for such a scenario?
4. (⋆⋆) Judge some sentences at `http://www.statmt.org/wmt08/judge/`. Afterwards, describe what are the most severe problems in bad translations:
   (a) missing words;
   (b) mistranslated words;
   (c) added words;
   (d) reordering errors;
   (e) ungrammatical output.
5. (⋆⋆⋆) Download human judgment data from a recent evaluation campaign, such as the ACL WMT 2008 shared task,[1] and carry out various statistical analyses of the data, such as:

[1] Available at `http://www.statmt.org/wmt08/results.html`

(a) Plot human judgment against BLEU score for all systems for a language pair and check for patterns (e.g., rule-based vs. statistical systems).
(b) Does intra-annotator agreement improve over time?
(c) Does intra-annotator agreement correlate with time spent on evaluations?