Evaluation Measures for Machine Translation: Rouge, NIST, WER, STM

Teslin Roys
University of Tübingen
1. Introduction and motivation

2. Similarity: Hamming distance example and approach

3. Cousins of BLEU: ROUGE, NIST

4. WER

5. STM

6. Summary and Discussion
Introduction and motivation
What is an evaluation measure?

- Any evaluation measure should in principle attempt to describe at least some aspects of a good translation.
- Here, we are looking for means of evaluation other than human judgement.
- This limits what we can easily evaluate:
  - What can we define clear, machine-understandable rules for?
  - What rules are computationally feasible?
  - Are the rules we choose too rigid?
Extrinsic vs Intrinsic

• An intrinsic evaluation measure looks at the output of a system and determines judges its general quality
• An extrinsic evaluation measures the effectivity of the output toward some function (e.g. task-dependent)
  • Is BLEU intrinsic or extrinsic?
Similarity: Hamming distance example and approach
Most of the measures we will see today use the following principle:

- “The closer a machine translation is to a professional human translation, the better it is” (Papineni et al., 2002)

For now let’s take that as a given and examine perhaps the simplest and most computationally efficient similarity measure available, Hamming distance.
Hamming distance

- Hamming distance is usually defined on bitstrings (i.e. binary numbers) but it functions on any string.
- If we have two bitstrings $a$ and $b$, then we can produce a bitstring $c = a \text{AND} b$.
- In every place where $a$ and $b$ match (e.g. both are 1’s, both are 0’s, both are a’s, etc.), $c$ is 1.
  - So a bitcount of $c$ gives us the number of matches.
- By comparing bitcounts, we can get a relative measure of similarity of a translation string to a reference for example.
- Because AND is a very cheap operation on any computer, we can do many, many comparisons of this kind.
• It may be cheap, but how good is this heuristic?
• On a character level, take the example a="flaw" and b="lawn"
  • They match in no place, yet have all the same characters
  • Hamming distance would say they have 0 similarity
• In what sort of situations would this measure fail in real world language examples? In what situations might it work relatively well?
Cousins of BLEU: ROUGE, NIST
• ROUGE or Recall-Oriented Understudy for Gisting Evaluation is a family of several measures
• Like BLEU, all the ROUGE measures compare with a reference text
• In fact, ROUGE-N is an overlap count of n-grams between the test and the reference text and almost identical (e.g. ROUGE-1 being a 1-gram count, for clarity)
• However, because the ROUGE measures were designed for use in both text summarization and machine translation, many of them penalize small differences (especially in length) to a lesser degree
• Reference: TV killed the radio star
• A. TV kill the radio star
• B. The radio star kill TV
• A = B because A and B both contain the same matching n-grams "TV", "the radio star"
ROUGE-L

- ROUGE-L is based on the notion of a longest common subsequence.
- To be more specific, we mean here longest subsequence of words (not necessarily contiguous) that we can find in the test and the reference.
- The intuition here being, the longer the LCS of two translations is, the more similar the two translations are. (Saggion et al. 2002, MEAD)
- A variation, ROUGE-W exists that weights consecutive matches more positively.
• Reference: TV killed the radio star
• A. TV kill the radio star
• B. The radio star kill TV
• A > B because A contains sequence ”TV the radio star” (4/5)
• Whereas B only has ”the radio star” (3/5)
• ROUGE-S counts matching skip-bigrams, meaning we ignore arbitrary gaps between words
• Good for long distance dependencies
• Very much like ROUGE-L, but we take every sequential pair instead of only the longest
• There also exists a variation which combines skip-bigrams with unigram co-occurrence statistics (ROUGE-SU)
• Discussion: Which is computationally cheaper, ROUGE-L or ROUGE-S?
• Reference: TV killed the radio
• A. TV kill the radio
• B. The radio kill TV
• C. The radio TV killed
• ROUGE-N: C > A = B
• ROUGE-L: A > B = C
• ROUGE-S:
  • A = 3/6 ("TV the", "TV radio", "the radio")
  • B = 1/6 ("the radio")
  • C = 2/6 ("the radio", "TV killed")
  • A > C > B
• Note: out of 6 because there are C(4,2) skip-bigrams in each sentence
• One of the more common variants of BLEU, the NIST metric, from the National Institute of Standards and Technology, is an alternate modification that adds scoring weights to each n-gram based on corpus occurrence statistics
• Namely, the less frequently an n-gram occurs in the corpus, it’s assumed it carries more informative value
  • This requires calculation of all n-gram frequencies in the corpus
• This intuition comes from the Shannon definition of information, which is measured in terms of the number of bits of information required to choose between two equally likely outcomes
• NIST also adjusts BLEU’s brevity penalty, weakening it so that small differences in translation lengths are penalized less severely
WER
WER, or the Word Error Rate, is a measure based on Levenshtein distance between the reference text and the hypothesis. For a whole corpus we define it as follows:

$$\text{WER} := \frac{1}{I^*} \sum_{k} \min_{r} d_L(E_k, \tilde{E}_{r,k})$$

- Here $I^*$ is the length of the reference (for normalization) and $d_L()$ is the Levenshtein distance.
- WER is commonly used in speech recognition evaluation as well as machine translation.
• Reference: TV killed the radio star
• A. TV kill the radio star
• B. The radio star kill TV
• In-class exercise: calculate the word-level Levenshtein distance of A and B from the reference and divide by the reference length.
One problem is that WER does not distinguish types of errors as being more or less significant: one could argue substitution errors are less recoverable than insertion and deletion errors. Different weightings have been proposed to address this. Because WER is based on Levenshtein, it is assumed to be at least $O(n^2)$ if certain theoretical tenets hold (see strong exponential time hypothesis).
STM
• In the previous measures discussed, the fundamental elements being compared are all variations of sequences or n-grams.

• Let’s take a look at another approach of Liu, Ding and Gildea (2005), incorporating syntactic structure.

• In general, first they parse the reference and test text with a PSG-style parser, then they compare the syntax trees rather than the text.
STM example comparison

Figure 1: Syntax Trees of the Examples
Sub Tree Metric (STM)

\[
STM = \frac{1}{D} \sum_{n=1}^{D} \frac{\sum_{t \in \text{subtrees}_n(hyp)} \text{count}_{\text{clip}}(t)}{\sum_{t \in \text{subtrees}_n(hyp)} \text{count}(t)}
\]

- In the above, \( D \) is the maximum depth of subtree considered.
- \( \text{count}(t) \) denotes the number of times subtree \( t \) appears in the hypothesis’ syntax tree.
- \( \text{count}_{\text{clip}} \) is the clipped number of times \( t \) appears in the references’ syntax trees, meaning the count from the hypothesis tree can not exceed the max number of times the subtree occurs in any reference’s tree.
One obvious issue is that D limits the depth of subtrees checked — but this can be solved by application of convolution kernels (see Liu, Ding and Gildea (2005)).

Another less tractable problem is the requirement to parse every sentence.

But the advantage lies in the ability to better handle translations like the following:

- Reference: I had a dog.
- A: I have the dog.
- B: A dog I had.
- Bigram based methods would normally find B > A which is unintuitive. With STM A > B because although the determiner and morphology is different, it is structurally alike to the reference.
Summary and Discussion
• The ROUGE family of measures vary in computational complexity from comparable to BLEU up to NP-hard, and are perhaps best applied in situations
  • Where the relative lengths of the hypothesis and any reference translation are less significant
  • And when capturing long-distance dependencies is important, ROUGE-S and ROUGE-SU are useful

• WER is based on Levenshtein, so roughly $O(n^2)$ where $n$ is the average length of reference, and is best applied in situations
  • Where the relative lengths of the hypothesis and any reference translation can be significant
Summary

- STM seems good to apply when
  - Word order and syntactical structure is more significant (more so than statistical co-occurrence of loose n-grams)
  - The cost of parsing every sentence is tenable
- NIST is essentially similar in time cost to BLEU, possibly with minor memory implications, and seems best applied when
  - Word order is not so significant
  - The matching of rare n-grams is more important than many n-gram matches
Discussion

• Is the goal to replace or supplant human judgement?
  • "The closer a machine translation is to a professional human translation, the better it is" (Papineni et al., 2002)

• Should we aim to make evaluation measures task and language-independent?

• Or are there advantages to specialization?
Questions?