CSCI 544: Applied Natural Language Processing

Assignment 7: Machine Translation Evaluation

Jonathan May (adapted from Chris Dyer)

Out: **10. November 2017**
Due: **1. December 2017**

Edit of 2017-11-02 16:36:11-07:00

Automatic evaluation is a key problem in machine translation. Suppose that we have two machine translation systems. On one sentence, system A outputs:
This type of zpisnku was very cenn writers and cestovateli.
And system B outputs:
This type of notebook was very prized by writers and travellers.
We suspect that system B is better, though we don't necessarily know that its translations of the words *zpisnku*, *cenn*, and *cestovateli* are correct. But suppose that we also have access to the following reference translation.
This type of notebook is said to be highly prized by writers and travellers.
We can easily judge that system B is better. **Your challenge is to write a program that makes this judgement automatically.**

Getting Started

The code we have provided includes a simple program that decides which of two machine translation outputs is better. Test it out!

```
python evaluate > eval.out
```

This script uses a very simple evaluation method. Given machine translations h_1 and h_2 and reference translation e, it computes $f(h_1, h_2, e)$ as follows, where $\ell(h, e)$ is the count of words in h that are also in e.

$$f(h_1, h_2, e) = \begin{cases} 1 & \text{if } \ell(h_1, e) > \ell(h_2, e) \\ 0 & \text{if } \ell(h_1, e) = \ell(h_2, e) \\ -1 & \text{if } \ell(h_1, e) < \ell(h_2, e) \end{cases}$$

where

$$\ell(h, e) = |h \cap e|$$

We can compare the results of this function with those of a human annotator who rated the same translations.

```
python compare-with-human-evaluation < eval.out
```

You should get an accuracy of 0.483.

The Challenge

Your challenge is to **improve the accuracy of automatic evaluation as much as possible.** Improving the metric to use the simple METEOR metric in place of $\ell(h, e)$ is sufficient to pass (with 80/100 provided you also upload reasonable code and explanation). Simple METEOR computes the harmonic mean of precision and recall with no chunk penalty or approximate matches. That is:

$$\ell_{\text{Simple METEOR}} = \frac{P(h, e) \cdot R(h, e)}{(1 - \alpha)R(h, e) + \alpha P(h, e)}$$

where P and R are precision and recall, defined as:

$$R(h, e) = \frac{|h \cap e|}{|e|}$$

$$P(h, e) = \frac{|h \cap e|}{|h|}$$

Be sure to tune the parameter α that balances precision and recall. This is a very simple baseline to implement and should elevate your accuracy to about 0.52, depending on how you tune. However, evaluation is not solved, and the goal of this assignment is for you to experiment with methods that yield improved predictions of relative translation accuracy, using what you have learned in this class. Some things that you might try (mouse over in the following list for links to relevant papers):

- Learn a classifier from the training data.
- Use WordNet to match synonyms.
- Compute string similarity using string subsequence kernels.
- Use an n-gram language model to better assess fluency.
- Develop a single-sentence variant of BLEU.
- Use a dependency parser to assess syntactic well-formedness.
- Develop a method to automatically assess semantic similarity.
- See what evaluation measures other people have implemented.

But the sky's the limit! Automatic evaluation is far from solved, and there are many different solutions you might invent. A whole series of papers came out of this work.

What you can and can't do (different for this assignment!)

You do not need any other data than what we provide but you are free to seek some out if you think it will help. Unlike with other assignments, for this assignment you are free to use **any code or software you like, except for those expressly intended to evaluate machine translation output.** You must write your own evaluation function. If you want to use part-of-speech taggers, syntactic or semantic parsers, machine learning libraries, thesauri, or any other off-the-shelf resources, go nuts. But evaluation software like BLEU, TER, METEOR, or their many variants are off-limits. You may of course inspect these systems if it helps you understand how they work. If you aren't sure whether something is permitted, ask us.

What to Turn In

- **(80 points; 60 for reaching at least the simple METEOR baseline on the full data set; more for exceeding it; final scores depend on other students' performance; reduced if machine translation evaluation code is explicitly used or score is below the simple METEOR baseline; no credit below the provided baseline)** Your automatic judgements of the dataset, uploaded to Vocareum. A leaderboard will show the same score you get when you evaluate locally, which is your performance on the first half of the data set. Scores on the second half will not be revealed until after the submission closes. Overfitting on the first half could lead to worse performance on the second half!
- **(10 points; reduced if code not uploaded or machine translation evaluation code is explicitly used)** Your code, uploaded to Vocareum, along with instructions on how to run it. This is for documentation purposes only.

- **(10 points; reduced if explanation is not clear)** A clear description of your algorithm and its motivation, to crowdmark. Note, this is *not* a list of things you tried that didn't work, or system results. The description does not have to be long. A fellow student reading this description should be able to reimplement your approach. If you are reimplementing an already existing idea from literature, be sure to credit the original work, in addition to your description of the algorithm.