Pun Identification and Extraction

Johnny Le, Jesus Zarate

January 26, 2018

1 Task

In our project we aim to detect and extract puns from a set of punning jokes. Puns belong to a class of language constructs, where the speaker or writer intends ambiguity on a certain word, allowing the word to carry two or more separate meanings simultaneously. Typically word sense disambiguation (WSD) assumes that for each word there exists a single unambiguous communicative intention.

There are two different types of puns, **homographic** puns and **heterographic** puns. Homographic or perfect puns are puns where the two meanings share the same pronunciation. On the other hand heterophonic or imperfect puns are those that rely on similar sounding signs, where the signs are considered as written rather than spoken sequences

2 Data

We will be using the SemEval-2017 Task 7 data. The data is in XML format with each pun joke listed sequentially.

Example of a pun joke:

```
<text id="hom_4">
<word id="hom_4_1">A</word>
<word id="hom_4_2">ditch</word>
<word id="hom_4_3">digger</word>
<word id="hom_4_4">was</word>
<word id="hom_4_5">entrenched</word>
<word id="hom_4_6">in</word>
<word id="hom_4_7">his</word>
<word id="hom_4_8">career</word>
<word id="hom_4_9">.</word>
</text>
```

There is also a text file that illustrates the answer format for the scoring program that came with the data. The format of the file consists of two fields separated by horizontal whitespace (a single tab or space character). The first field is the ID of a text from the XML field. The second field is the ID of the one word in that text which is a pun.

Example answer for the pun joke above from the text file:

hom_4 hom_4_5

3 Evaluation

Since the data came from SemEval-2017 Task 7 which contains a scoring program, and it uses the F-measure statistic.

We will be implementing our evaluation program using python. We will compute the following:

Recall

The number of correct puns (location of the pun word) extracted divided by the number of puns in the gold file.

Precision

The precision will be computed by taking the number of **correct** puns extracted divided by the total number of puns extracted by our system.

F-measure

The F-measure is how we're going to be measuring the performance of our extraction system.

$$F_1 = 2 * \frac{Precision * Recall}{Precision + Recall}$$

The data comes with two sets of data a training data, (which we will do the training with) and test set.

The training data will be split into two sets, one for training and the rest for test and then tune. Finally the test data will be used to evaluate the system.

4 Demo Interface

The demo interface will consist of running a python script on the command-line.

python extract.py input-file.txt

Executing the command above will yield two files. The first file identified pun word in each text, and the second file will be the evaluation results discussed in the previous section.

subtask1-heterographic-test.results
subtask1-heterographic-test.score