Statistical Machine Translation

Exercise 1

To be sent (pdf, Zip) to `waszczuk@phil.hhu.de` by 08.11.2018.

Exercise 1 - Practice

This week we will start using Java to extract information from a parallel corpus. Download the code of the exercise (Zip file) on the course webpage and save it locally. The code is made available as an Eclipse Project, so you just need to start Eclipse and import the archive as existing project.

At the end of the session, export your project as Zip file and save it for the next sessions (Email, USB).

Add a new `Run Configuration`. The `Main` class is `de.hhu.phil.smt.misc.Uebung1`. The arguments of the program are `data/train.de.h250.lc data/train.en.h250.lc`. The code should now be compilable and executable. As an output, you can read the number of parallel sentences. You can start by getting familiar with the code in the different files.

(a)

Your exercice is to count how many different german-english wordpairs are included in our parallel corpus `train`, consisting of 250 sentence pairs. You should build the following set:

$$\{(f_j^s, e_i^s) \ : \ j \leq |\boldsymbol{f}_s|, i \leq |\boldsymbol{e}_s|, 1 \leq s \leq 250\}$$

f_j^s is here the jth word in the sth sentence $\boldsymbol{f}_s$ in the source language, here german, and e_i^s is the ith word in the corresponding english sentence $\boldsymbol{e}_s$. The output of the program is the cardinality of the set. If you want to check the set which you built, you can also print its elements.

The entry point is the `main`-method in `Uebung1.java`. You mainly have to add code in the `TransCooc.java` file. You can use the class `Pair`, already implemented, as well as other classes of the Java API (`http://docs.oracle.com/javase/8/docs/api/`), for

instance Sets, Maps (associative tables)....

Info: The set computed by the program represents all possible word alignments in the corpus. Concretely, this permits to answer this kind of questions: into which word could the third word in the first German sentence be translated? The first/second/third.../last word in the first English sentence are possible candidates. When training for word alignment, the best overall alignment must be found.

(b)

Word alignments should not be limited to 1-1 relations, but also allow more words from the source language (German), for example two. To do so, build the following set:

$$\{((f^s_{j_1}, f^s_{j_2}), e^s_i) \ : \ j_1, j_2 \leq |\boldsymbol{f}_s|, j_1 < j_2, i \leq |\boldsymbol{e}_s|, 1 \leq s \leq 250\}$$

Please print the size of the created structure.

General note: You do not have to stick exactly to the proposed structure of the code. If you are not sure, feel free to ask, but in any case, make sure that anyone can understand your code, for example by using comments. If the original entry point of the program no longer works, please add a readme file that specifies how the code should be compiled and executed.

Exercise 2 - Theory

There is a disease that strikes one person out of 100,000 people. A test has been created in order to diagnose the disease. The test yields a positive result (ie, indicates that the subject has the disease) with a probability of 0.98 when the subject is ill. When the subject is healthy, a positive result comes with a probability of 0.007.

Bob takes this test and the result is positive. What is the probability of Bob actually being sick?