Statistical Machine Translation

Homework 6

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

Exercise 1 - Practice

This week, we will implement a very simple **Decoder** for phrase-based translation. Download the code (Eclipse project as a zip file). Start Eclipse and import the zip file as existing project. At the end of the session, you can export your project as zip archive and keep a copy of it (email, USB).

The models which will be used to weight the translation are (1) the phrase translation probability (2) the Bigram language model probability and (3) distance based costs for the re-ordering of the phrases in the translation (see the script about phrase based translation). We already implemented a Bigram language model: `de.hhu.phil.smt.lm.BigramModel`. Implementations for the re-ordering model and for the translation model (based on the phrase pairs that we have extracted in Übung 5) are available: `de.hhu.phil.smt.pb.tm.ReorderingModel` and `de.hhu.phil.smt.pb.tm.TranslationModel`. You will need this models for decoding, but they can be seen as black boxes.

The main class in the exercise is `de.hhu.phil.smt.pb.de.Uebung6`. It compiles, but the four test sentences are not yet translated. Output in the end the best translation given by your decoder.

`de.hhu.phil.smt.pb.dec.Hypothesis` represents a hypothesis, i.e. a node in the search tree/graph. For the decoding, this class is fundamental. Implement the two methods `isComplete()` and `isUncovered(j1, j2)`.

Now the main task: Implement the method `decode()` in `de.hhu.phil.smt.pb.dec.Decoder`. Some functions are already implemented. Tip: First, explore the whole search tree (this is possible with the sample data). Then, you need make the recombination of hypotheses possible. Indicate how many complete hypotheses can be found for each sentence (you can also print the output of these complete hypotheses together with their weights).

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.