Lehrstuhl für Informatik 6
Rheinisch–Westfälische Technische Hochschule Aachen
Prof. Dr.–Ing. H. Ney

July 7, 2015
Andreas Guta

11. Exercise Sheet **Statistical Methods in Natural Language Processing**

The solutions to the problems may be submitted until **Tuesday, July 14, 2015** before the exercise lesson. Upload a digital version to the L2P and bring your written/printed solution to the exercise lesson. Condition for obtaining the **Leistungsnachweis** (*Schein*) "Statistical Methods in Natural Language Processing" is the successful solution of 50% of the problems and the presentation of the solution of at least two problems in the exercise lessons.
The solutions to the problems can be submitted in groups of up to **two** students.
For programming exercises:

- The implementation has to be done in C++, but you can use additional standard Unix tools (tr, sed, awk...) for the preprocessing.
- Implementations have to be uploaded to L2P as well as printed out and submitted before the exercise lesson.
- Upload the programs as a `.tgz` or a `.zip` compressed directory to L2P. It must include a `Makefile` and must compile on Linux using simply the `make` command.
- Include a *short* description of the main data structures and algorithms used in your solution sheet.

1. [3 Points] Implement (efficiently) the WER and PER measures for evaluation of machine translation.

   - WER (word error rate): the minimum number of insertions, deletions and substitutions divided by the number of words in the reference.
   - PER (position-independent error rate): word error rate which does not take into account the order of the words.

2. [7^+ Points] Implement the DP search for monotone alignments as presented in the lecture. You can find some already trained data on our webpage. It consists of:

   - Monotone alignment probabilities derived from the HMM alignment probabilites as calculated by (a modified version of) the publicly available toolkit GIZA++[1]. The training sequence was 5 iterations of Model 1 and 5 iterations of HMM.
   - Lexicon probabilities extracted from the same training.

   In the `README` file included in the package you will find a description of the format of the files. What you still have to compute:

   - A (bigram) language model. You may reuse the implementation of exercise 5.
   - A length model. Use the pooled model $\lambda_I = \lambda \cdot I$.

   Translate the given test corpus and compute the WER and PER using your above implementation. Do you see any systematic errors?

[1] See `http://fjoch.com/GIZA++.html`.