OPDRACHT 2
DISTRIBUTIONAL SIMILARITY
TEAM ASSIGNMENT

1. Download the corpus (the first part of UkWAC, UKWAC-1.xml.sent) for measuring similarity and the list of word pairs. Training data is to be found in the file **discrimination_train.tbl**, test data in **discrimination_test.tbl**. Read the description on the free associations task from

   `http://wordspace.collocations.de/doku.php/data:esslli2008:correlation_with_free_association_norms`,

   section 1 (Discrimination) from *Data sets and Tasks*.

2. Define the context (using lemma and the ± 2 tokens, ± 3 tokens around the words taken from input data) and extract frequency data. Note that UKWAC-1.xml.sent contains sentences with lingustic information:

   ```
   word\tPoStag\tlemma
   ```

To extract only lemmas, one can use a perl one-liner like

```
perl -pe 's/\t*\S+\t\S+\t(\S+)/$1 /g' UKWAC-1.xml.sent  |
perl -pe 's/\<\/*s\>//g' > UKWAC-1.xml.sent.lemma
```

3. Implement the following similarity measures: cosine and Dice, and then the PMI association measure and frequency alone to build a co-occurrence matrix.
4. Run these implementations on the pairs from **discrimination_train.tbl** and, separately, on **discrimination_test.tbl**.
5. Find thresholds for all three classes (FIRST, HAPAX, RANDOM) on the training set either using exhaustive search or, alternatively, a linear classifier from WEKA (`http://www.cs.waikato.ac.nz/ml/weka/`).
6. Use the thresholds found in the previous step to classify test data from **discrimination_test.tbl**. Write down accuracy, precision, recall and F-score per class and for all three classes.
7. Analyze the results. Which measure leads to the best performance?

For an extra point:

1. Vary the length of the context - does it affect the results?
2. Experiment with a cut-off parameter (>1, >5). How does it affect the final performance?

3. Compare your results against those by Wandmacher et al. `http://ikw.uni-osnabrueck.de/~eovchinn/papers/ESSLLI'08_WOA_final.pdf`.

4. Experiment with additional information found in the UkWAC (e.g., filtering information in contexts based on the PoS data).

Tips:

1. To extract lemmata, run (double-check if the first line in the script corresponds to where **sh** can be found on your computer, **which sh** - if the folder is different, edit the first line):

```
chmod +x test_lemma.sh
./test_lemma.sh yourcorpus
```

   the output will be stored in a file **yourcorpus.lemma**

2. To extract tokens in the window of ± 2 around the word **day** you may use the following pipeline of commands (on Unix-like systems, from the command line):

```
chmod +x test.sh
./test.sh yourcorpus.lemma
```

   the output will be stored in a file **day.freq**.

3. An example of measuring cosine similarity between **cat** and **dog** (one of the measures that has to be implemented) on frequency counts:

	feed	white	bark
cat	10	5	0
dog	8	5	10

$$cosine(cat, dog) = \frac{10 * 8 + 5 * 5 + 0 * 10}{\sqrt{10^2 + 5^2 + 0^2}\sqrt{8^2 + 5^2 + 10^2}} \quad (1)$$