Lecture
"Foundations of Language Science and Technology: Statistical Natural Language Processing"

Prof. Dr. D. Klakow

Exercise

"Spam, Spam, Spam!" Unsolicited email is commonly referred to as "spam". Your task for this exercise session is to build a Naïve Bayes classifier that can automatically classify a given document (e.g. incoming email) using the two classes `SPAM` and `HAM`(=not spam).

Subtasks:

1. Download the data from

   `https://www.lsv.uni-saarland.de/fileadmin/teaching/foundations/ws1415/spam_dat.tar`

   The package `spam_dat.tar` contains training and testing data for both spam and non-spam messages.
   You can unpack the package with:

   ```
   tar -xf spam_dat.tar
   ```

2. Build a word-based language model for each of the classes, `HAM` and `SPAM`. The training data are in the files `ham_training` and `spam_training`, respectively. Instead of using relative frequencies as your probabilities, smooth your language models with *absolute discounting*. Use the following equation:

   $$P_d(x_i|\omega_k) = \frac{\max(N(x_i,\omega_k)-d,0)}{\sum_{x_i} N(x_i,\omega_k)} + \frac{d \cdot n_+}{\sum_{x_i} N(x_i,\omega_k)} \cdot \frac{1}{|V|}$$

   where $n_+ = \sum_{x_i:N(x_i,\omega_k)>0} 1$. It would be nice if the discounting parameter d could be optionally specified as a command line option. Use $d = 0.7$ as a default value. The file `vocab_100000.wl` contains the vocabulary with which you are to build your language models.

3. Write a classifier that classifies each mail contained in the test file `ham_spam_testing`. A new mail begins with `#*#*#` followed by the class label. Your program should output the predicted class for each mail in the test file along with the overall number of correctly and incorrectly classified mails. The predicted class for the test mail is the class which has the highest probability according to the class-specific language model and the class prior. In order to prevent an underflow in your classifier use a sum of logarithms instead of a product of the probabilities:

$$\hat{\omega} := \arg\max_{\omega_k} \log P_d(x|\omega_k) + \log P(\omega_k)$$

where $P_d(x|\omega_k)$ is a class-specific language model and $P(\omega_k)$ is a class prior.

General Remarks:

- You may implement this exercise in any programming language. However, we recommend either of the following languages: *Perl*, *Python* or *Ruby*.
- Please document your code properly, so that it can be easily tested without reading the entire source code.
- We will start discussing the exercise already on Friday the 5th with a question/answer session on problems that your have encountered up to that point. Monday December 8th will also be Q/A.
- The final **deadline** for this exercise is 9/12/2014 11am. Earlier submissions are welcome.
- Send your program to `Michael.Wiegand@LSV.Uni-Saarland.De` and `Dietrich.Klakow@LSV.Uni-Saarland.De`
- In case you have further questions with regard to this exercise, please send your mail to Michael Wiegand and Dietrich Klakow.