Speech and Language Processing 1 - Assignment 2
Gender detection in social media files

Deadline: September 28, 2015
Questions? Rieks op den Akker (h.j.a.opdenakker@utwente.nl), Gwenn Englebienne(g.englebienne@utwente.nl),

The language use of a person depends on many factors, such as the person's age, gender, social class etc. In this assignment, we will look into differences of language use between males and females. We will build statistical models that we then use to predict the gender of a person based on text she/he has written. Being able to automatically classify the gender based on a person's text has many practical applications. For example, researchers that are analyzing trends on the web are interested in differences between certain user groups (based on gender, age, etc.).

In this assignment we will use statistical language models to model the language use of male and female persons, and to automatically classify new text into the male or female class.
The method that we will use is known as "Naive Bayes". It is an application of probability theory and uses Bayes' rule. (see your course book)
You are free to use any programming language you like.

1 Manual classification

To get started, for both examples, guess the gender of the person who has written the text, and provide a short motivation.

> **Person 1** Yo dudes. Just wanted to see what was up for Monday night. I'm thinking that perhaps we can chizzill at Bri-Guys, since he said he'd prolly like to hang out [...]
>
> **Person 2** yeah, graduated, go me. went bowling afterwards, i was bored cause i was too tired to do anything really. my mom stole my energy drink.

<u>**Submit**</u> Your guesses & motivation.

2 Dataset

In this assignment you should **choose one of the following datasets**.

- **Twitter** dataset. A collection of persons and their latest tweets. This dataset is in Dutch.
- **Blogs** dataset. A collection of persons and their blog posts. This dataset is in English.

The input directory contains two subdirectories:

- *Train* These documents will be used to train your language model. (600 docs)
- *Test* These documents will be used to test your model. (50 docs)

The documents are named as [gender]-[person ID].txt. There is also a file called groundtruth.txt, this file contains the correct labels for the documents in *test*.

3 Tokenization

The first step is to tokenize the data. Tokenization splits up a character sequence into smaller pieces (tokens). You also want to normalize your tokens (for example by converting everything to lower case). An example tokenization is (sentence from blog corpus):

> **Original sentence** Hello, everyone. Do you like the new layout?
>
> **Tokens:** [Hello, everyone, Do, you, like, the, new, layout]
>
> **Tokens normalized:** [hello, everyone, do, you, like, the, new, layout]

For this assignment, make a *simple* tokenizer. Since our datasets are small, we recommend to make strong normalizations, for example by removing punctuation.

Submit Provide a description of how your tokenizer works. Select 3 sentences, show the original sentence, and the tokens you obtain using your tokenizer.

4 Vocabulary

First, answer the following questions using the documents in the *train* directory:

Submit

- How many unique n-grams are there? (where n=1,2,3).
- Report the top 10 most frequent words (= unigram) and their frequencies.
- How many words occur 1,2,3,4 times in the corpus?

For the rest of the assignment, we will only work with *unigrams*. Run the tokenizer on all documents in the *train* directory and keep track of the word frequencies. Keep the words that occur at least 25 times as your vocabulary. Modify your code such that all words that are not in your vocabulary are ignored in the rest of this assignment.

5 Text classification using a unigram language model

Recall that for a text with words w_1 ... w_n, we calculate the probability as follows using a unigram language model:

$$P(w_1, w_2, ..., w_n) = P(w_1)P(w_2)...P(w_n) = \prod_{i=1}^{n} P(w_i)$$

In order to avoid underflow, this is usually calculated in log space (base 2):

$$logP(w_1, w_2, ..., w_n) = log\prod_{i=1}^{n} P(w_i) = \sum_{i=1}^{n} logP(w_i)$$

In our dataset we have two classes: *male* (M) and *female* (F). For each class, we will calculate a separate language model. This is the *training* or *learning* phase. In the apply phase, we will classify new texts as written by *male* or *female*. For *testing* our machine learning classifier we apply the models on the documents in the test part of the corpus.

1. **TRAIN**. For the documents in the *training* directory, build two language models. One using the documents written by *females*, and one using the documents written by *males*.

   For example, we calculate the probability for the *male* language model as follows.

   $$P(w_1, w_2, ..., w_n|M) = \prod_{i=1}^{n} P(w_i|M)$$

   Where we are using the conditional probability ($P(w_i|M)$ instead of just $P(w_i)$), because we are calculating the probabilities using only the documents written by males. We estimate the conditional probabilities:

   $$P(w_i|M) \approx \frac{C(w_i, M)}{N_M}$$

   where $C(w_i, M)$ is the frequency of word w_i in the documents written by males and N_M the total number of words in the documents written by males.

2. **TEST**. For the documents in the *test* directory, calculate the probability for both language models. Assign each document the class for which it has the highest probability. Using the MAP (Maximum Aposteriori Probability) rule:

   $$Class(D) = argmax_C P(C|D)$$

Smoothing Use smoothing to avoid zero probabilities:

$$P(w_i|M) \approx \frac{C(w_i|M) + k}{N_M + kV}$$

Where V is the size of your vocabulary. Report the results for two settings: when $k = 1$ and a value for k that you have selected yourself. Although this is a simple method, it is often used for text classification, in particular in combination with the classification method that we are using.

Evaluation Provided is a python script that can be run as follows:

```
python evaluate.py groundtruth.txt [results file]
```

This will print the accuracy of your predictions on the documents in the test set.
It requires as input on each line [documentID] tab [M/F]

Submit The performance of your classifier (accuracy) for both runs (smoothing with $k = 1$, and a k that you have selected yourself).

Optional background information: The method presented here is in fact the same as a multinomial naive Bayes classifier, with equal prior class probabilities. In our case, this makes sense, since we expect the proportion of males and females to be (almost) equal. In case this is not realistic we can estimate the prior class probabilities $P(Class = M)$ and $P(Class = F)$ from the corpus, using maximum likelihood estimation.

6 Characteristic words

We will now analyze which words are highly characteristic for either women or men (in our dataset):

- Rank words according to $\frac{P(w_i|F)}{P(w_i|M)}$ (this will rank words that will cause the model to believe a text is written by a female).
- Rank words according to $\frac{P(w_i|M)}{P(w_i|F)}$ (characteristic words for males).

You can experiment with the vocabulary (for example by selecting a higher or lower threshold of words to include). You'll probably want to set a reasonably high threshold (e.g. min. occurence of 50) to prevent rare words to appear at the top.

Submit Look at words that are ranked high. What kind of words are ranked high? Do you think they make sense? Report the top 10 words characteristic for males, and the top 10 characteristic for females.

7 Error analysis

Analyze a document that was wrongly classified.

Submit Provide the document ID and a short explanation of why you think the model classified the document incorrectly.

8 Improving the classifier

Discuss ways how you can improve the classifier. Are there other types of features that could improve the classification? What are the disadvantanges of the current way of normalizing and preprocessing the text, suggest possible changes. Motivate your suggestions, for example based on observations during the error analysis.

Submit Suggestions for improvements and your motivation.

9 What to submit

- Put your answers to the questions above in one document.
- Clearly state your name(s) and the dataset you used on top of the document.
- Put the document and your code (not the data files!) in one zip file.
- Use your names in the name of the zip file.
- Submit the zip file on blackboard