Assignment 4

Remarks:

1. Please submit your work until **December, 05 15:00** on Ilias.
2. If you need any **help**, send an email to `max.kisselew@ims.uni-stuttgart.de` or drop in at my office: 1.013 (Pfaffenwaldring 5b).

Exercise 1 (IIR 8) [3 P.]

Sentiment analysis "aims to determine the attitude of a speaker or a writer with respect to some topic or the overall contextual polarity of a document"[1]. Assume, the IMS develops a novel approach for sentiment analysis. To test the success of the new approach a gold standard is needed. Thus, two annotators independently annotate 200 documents regarding whether they convey a positive attitude or not. The following table shows how often they agreed. Calculate the Kappa coefficient for the agreement between the two annotators. Will it be possible to construct a gold standard from this annotated data?

		Judge 2: Document positive?		
		Yes	No	Total
Judge 1:	Yes	120	30	150
Document positive?	No	40	10	50
	Total	160	40	200

Exercise 2 (IIR 14) [$x \geq 7$ P.]

Develop a simple kNN classifier which asks the user for a k and then assigns a class to new documents. Use either Java or Python. Consult me if you would like to use another programming language. Proceed step by step:

1. On the course homepage you can download a corpus consisting of 2487 documents. The documents are labelled either "spam" or "ham". The first line of each document contains its class and the following lines contain the text of the document, e.g.:

```
ham
Hi to all
Here is my problem . I'm trying to export this data to SQL database .
[...]
```

2. Implement a method/function that reads in the documents and creates their document vectors. Omit the stop words listed in `stopwords.txt` (on the course homepage). Use the *tf-idf* weighting for the vector values. Normalize the vectors to unit vectors. Keep in mind to save the information about the class of a document while it is processed.
3. Write a method to calculate the Euclidean distance between two document vectors.
4. Choose randomly 10 documents from the corpus which will serve the classifier as a training set. 5 documents should be of class "spam" and the other 5 of the class "ham". Put the training set documents into a subdirectory named `train`.

[1] `http://en.wikipedia.org/wiki/Sentiment_analysis`

5. The main program now should be able to take the number of next neighbours (k) the classificator should consider when assigning the test set documents to a class. Then it should perform kNN classification with the given k on a randomly chosen test document from the corpus and print out the following information:
   - Which class has been assigned to the test document?
   - Is it the correct class?
   - Which documents from the training set have been responsible for the decision (i.e. which ones had the smallest distance to the new document)?
6. Your program must be executed with three arguments:
   the folder containing the training files, the test file and k. For example:
   `python kNN_classifier train test_file_1 3`
7. Test your classifier with several documents from the corpus. Keep in mind that the test documents may not occur in the training set. Now train your classifier on 2000 documents. Compare the amount of correctly classified test documents. Does the larger training set improve the classification results?
8. You are free to choose the structure of the program code and method/function names and parameters. But please follow the requirements above.

Due date: Wednesday, December 05, 2012, 15:00