CIT 653: Practical Data Mining
Spring 2018
Prof. Dr. Samhaa R. El-Beltagy
Due Date: 13/05/2018

Nile University

Term Mini Project

For this project, you may work in groups of up to **4 members**.

Description

Write a text classification pipeline to classify tweets as either: positive, negative or neutral. You have the choice of working with either English or Arabic. A dataset (training and testing) will be provided for each.

To carry out your project, you will be provided with the following resources:

1) A sentiment lexicon.
2) A set of Arabic stopwords. You will probably find other sets online.
3) A paper describing each of the provided datasets

You are completely free to use one or more of these resources or none of them. You can use any resources that are available on the internet and you can create your own resources (such as extra training data).

Assuming you follow a supervised approach, running the data through a classifier, is probably the simplest part of the project. Different parameters and configurations will often result in increased accuracy, but what I really want to see is you being creative about the features and how to use them. You MUST introduce at least three new features (besides the document vector) and show how each affects the results. At the end, you should only keep features that resulted in improved performance. I strongly advise you to include in the project documentation, any new feature that you have introduced, even if it decreased the accuracy, as that will show me the effort you have put in the project.

Step 1:
Do minimum pre-processing on the data just to be able to run it through a classifier. We call this processing of raw data. Results from this step will provide you with your baseline. Run the data through various classifiers and decide on which one you will use. Report these results in your documentation as your "baseline". Please use 10 fold cross validation when reporting your results on training data. The baseline results are the results that you start with and aim to improve. Your goal from this step onwards will be to improve over the baseline in terms of **both** accuracy and F-score.

Step 2:
After you are happy with the system you have built, you will apply it on unseen test data that will be provided at a later date. You will be asked to upload the results of applying your model to" Kaggle in class" https://inclass.kaggle.com/. Details regarding this step will also be provided later. Once you upload your results, they will be automatically evaluated and you will see how your results compare to those of other teams. A bonus will be given to the highest scoring team. If you are not happy with the performance of your system, you can always go back, modify it and re-submit your results.

What to turn in:

1) Project documentation (details given below)
2) If you are using Weka, the .arff file that produced the best results (or the data file you used if you are using a tool other than Weka)
3) Your source code

Project documentation:

Please make sure that the documentation is easy to read and follow. Python output that is presented in the form of a screen dump with no formatting is not acceptable. Do not include code in the documentation. You must provide enough details so that the reader of your document can understand what you are trying to do and re-produce your results or something close to those results. Statements like "I've used a Weka filter", without specifying which filter you are referring to, are not acceptable. It will even be better, if you abstract away from implementation details and stick to the technicalities, so for example, if you've used an attribute selection filter configured to carry out feature selection using information gain, you can simply state that you used information gain for feature selection. The structure of your document should be as follows:

- Introduction: State the problem and what you are trying to accomplish
- Data description: Provide a description of the used dataset (number of instances, features, division between training and testing, etc).
- Baseline experiments: Here you must state what the goal of the experiments is and after you have presented those experiments, what your conclusion is.
- Other experiments: each experiment must have a goal, a set of steps, results and a conclusion.
- Overall conclusion
- Task distribution: This should describe how the tasks have been divided amongst team members (I need to know who did what).

You documentation should also include:

- A list of tools that you have used (you may use any tool or programming language you are comfortable with)
- A list of any external resources that you have used

When including tables, make sure they have a caption that illustrates what you are presenting, if its f-score than state, if its accuracy than also state.

Since the datasets you are using have been benchmarked, compare your result to previously obtained ones.

After you are done with your experiments answer the following questions:

1) What was the biggest challenge you faced when carrying out this project?
2) What do you think you have learned from the project?

What to turn in:

1) Project documentation
2) If you are using Weka, the .arff file that produced the best results (or the data file you used if you are using a tool other than Weka)
3) If you are using Python or any other programming language, your source code.

The breakdown for project grade is given below:

Item	Points
Baseline Experiments	/10
Data-Cleaning and Pre-processing	/15
Ability to approach the problem in a sound and creative manner	/15
Inclusion of additional features	/20
Submission of results to Kaggle in-class	/15
Providing required data/source files	/10
Following documentation guidelines	/15
Bonuses	0
Late Penalty	0
Total	/100