Abstract

This document provides a template for your final project report. It illustrates the sections you need to include in your report. The final report should be submitted as a PDF file. it should be **no more than 2,500 words long**. Keep the discussion focused and to the point, shorter is better. Clear and complete write-ups with strong discussion and analysis sections will receive up to 25 points extra credit for the final project.
All writing should be in your own words, no copying of material written by someone else is allowed.

1 Introduction (15 points)

In the introduction, give a general overview of your approach. Explain what intuitions you had about the task—what types of features would be applicable and helpful. Also identify 2-3 ideas you would like to test: for example you may think a certain type of feature would be more suitable than others.. Then describe the general conclusion about your hypothesis: did it turn out the way you expected, according to the cross-validation and the five test results you ran. Include any general thoughts about what you learned about the task as a result of your work.

This should be a general description of what you hope to achieve and what you found out, not details.

This section will be evaluated entirely based on the clarity of your description.

You can assume your reader is the course instructor or in general an expert in NLP.

Note: if you chose the task for your project, the general description of the task would also appear here. We are skipping this aspect because you are all working on the same task.

2 Method (10+15 points)

A good system may either relay on a number of features to improve the prediction accuracy or it could rely on your understanding of the task, for example changing the subset of the training data you use or how you use the training data, i.e. what classifier you run. You should definitely have a subsection on features. The section of other improvements is optional and can be omitted if it is not relevant to your approach.

2.1 Features

In this section, describe in detail what specific features you decided to implement, what tools you used to generate the features and what new functions you implemented yourself.

2.2 Other improvements

Changing in machine learning approach, use of data, and any similar idea for improving the performance on the task.

2.3 Resources and tools

Describe the resources or tools (i.e. Wordnet, Stanford Parser, MPQA) you used and why they were needed

Half of the points (10) will be based on the clarity of your description and on how conniving (practical or based on knowledge we have learned in class) is your argumentation for using features. The other half would reward the implementation and ability to use other tools and resources and the ability to convincingly argue the use of a particular type of feature or approahc.

3 Your final system (5+20 points)

Here give the name of your team name for the leaderboard and give an overview of the system. No need to repeat information presented in detail in the previous section. Please make sure that you describe the system that corresponds to your last submission to the leaderboard. We would expect that the code you submitted will match the workflow description presented here.

You may use flowchart, graphics or pseudo-code to demonstrate your algorithm/system, or simply explain in text.

Most (15 out of 20) of this section will be scored based on the implementation you submitted for the project. The rest of the points will reflect the clarity of description.

4 Experiments (10 points)

4.1 Final submission system

Report the performance of your final system on the training set (optional) and the test set (required). You need to perform k-fold cross validation in order to get

the accuracy on the training set.[1] You can use the leaderboard to get the test set accuracy.

4.2 Other system alternatives

In addition, briefly describe what other settings of the system you tested on the leaderboard, what question about the performance of you system prompted you to evaluate that particular setting and what did you expect the result would be.

Accuracy	Train acc.	Test acc.
Final submission	*.*	*.*
Other submissions	*.*	*.*

Table 1: Performance of the final system.

”Other submissions” can be any other set of features, ablation experiments (all your features without the lexical features for example, to test if fewer features will still give you competitive results) or any experiments designed to test of certain intuitions about the task hold true.

In this last class of features, the performance may end up being low, much lower than the other systems you see on the leaderboard. This is ok. For these experiments, having a performance that beats chance is a victory, especially you have an intuitive class of a few features that achieve that performance. A trivial example for an experiment that could fall in this category is for example to say you think information-dense texts will more and different named entities than non-informative texts. Then train a system that uses only a handful of features related to named entities. How about a system that simply looks at the number of capital letters and number of sentences that have more than one capital letter (which will be way faster than performing NER). You get the idea.

5 Discussion (25 points)

Discuss what have you learned about the task and your ability to solve it, based on the experiments you performed. What worked, what didn't work, what worked as expected, what surprised you.

Discuss any **special aspects of your system** that may have made it particularly successful.

[1]This article describes how to do k-fold cross validation: `http://www.csie.ntu.edu.tw/~b92109/course/Machine%20Learning/Cross-Validation.pdf`

For each conclusion, point to the experimental results that allowed you to draw the conclusion.