CS 6320.002: Natural Language Processing
Fall 2019

Project Milestone 4 – 80 Points
Due 8:30am 09 Dec. 2019

Deliverables: A tarball containing the following (submit just one per team):

- A PDF final report. Put the names of ALL team members on your report.
- Python files implementing your improved system system.
- A trained improved model, `improved.model`.
- A Python file `evaluate.py`.
- The test set for your project.

1 Improvements – 30 points

For this final milestone, your team must implement improvements over the baseline system, one improvement per team member.

What counts as an improvement?

- New types of features (at least two). For example, you might add scores from a lexicon that previous work did not use; that would count as one new type of feature. Switching from vanilla n-grams to skip-grams would be another new type of feature.
- New model. For example, switching from a rule-based to a statistical approach, or from a non-neural to a neural approach.
- New training scheme. For example, switching from fully-supervised to semi-supervised, or from supervised to unsupervised.
- New loss function. For example. switching from cross-entropy loss to hinge loss or reinforcement learning.

If you're not sure whether or not something counts as an improvement, ask! You can choose multiple improvements from the same category. For example, you can do six new types of features as two improvements.

For each improvement, you must be able to justify why you think that improvement will increase the performance of your system. See the Final Report section for details.

As with Milestone 3, you can use as many files as you want to organize the code for your system. You may use machine learning or NLP toolkits (eg. SciKit-Learn, Spacy, NLTK) to for preprocessing and models, but you must code the rest of the system from scratch. You may NOT submit existing code that implements the system (eg. that you found on a GitHub repo).

2 Experiments – 10 points

For each improvement, train and evaluate a new model that includes just that improvement. For example, if you have three team members, you must have three improvements, A, B, and C. Train and evaluate baseline + A, baseline + B, and baseline + C separately.

Then, if you have multiple improvements (ie. you are not a solo team), train and evaluate models for each combination of improvements. For example, if you have three team members, you should train and evaluate baseline + A + B, baseline + A + C, baseline + B + C, and baseline + A + B + C. If some of your improvements are mutually incompatible (eg. if one improvement is an unsupervised approach and the other is a semi-supervised approach, you can't use both at the same time), then skip that combination.

Save your best-performing model from the previous section in a file named `improved.model` and submit it. As with Milestone 3, you do not need to submit the training data, only the trained model.

3 Evaluation – 5 points

Submit a file `evaluate.py` that does the following:

- Load your best-performing improved model from file.
- Load the test data.
- Make predictions/generations for the test data using your trained model.
- Evaluate the predictions/generations.
- Print the metric scores.

This evaluation script may be identical to the one you submitted for Milestone 3, or it might require slight modifications (eg. if you changed learning algorithms or need to extract new features).

4 Final Report – 35 points

Your final report should follow this formatting template:
`http://acl2020.org/downloads/acl2020-templates.zip`

You should have the following sections (all suggested lengths are roughly estimated):

4.1 Abstract – 5 points

1 paragraph (100 words) summarizing your project task, approach, and results.

4.2 Introduction – 2 points

3-5 paragraphs (300-500 words) about your project task. Describe what the task is and why it is important. Give example applications and an example of what input and output

should look like for your system. You can probably reuse a lot of what you wrote for your project proposal for this section.

4.3 Related Work – 2 points

3-5 paragraphs (300-500 words) summarizing previous or related work on your task. Discuss 3 or more papers; describe their data, approaches, and results, and how they relate to your system. You can probably reuse a lot of what you wrote for Milestone 1 for this section.

4.4 Data – 1 point

1-2 paragraphs (100-200 words) describing the dataset(s) you are using. What do the input/output pairs look like, how many are there, where did the annotations come from, etc. You can probably reuse a lot of what you wrote for Milestone 2 for this section.

4.5 Methodology – 10 points

1-2 paragraphs (100-200 words) describing your baseline system. What model, learning scheme, hyperparameters, etc. did you use? You can probably reuse a lot of what you wrote for Milestone 3.

Then, 1-2 paragraphs (100-200 words) describing *each* of your improvements. What is the improvement and how does it work? Why do you think it should help for your task? What flaw or weakness in the baseline system is it meant to address? How did you implement the improvement? If someone else wanted to reimplement your system, they should be able to figure out how to implement your improvement based on your description alone.

4.6 Experiments – 10 points

1 table showing the performance of your baseline and each of your improvement experiments (see Experiments section).

Then, 3-5 paragraphs (300-500 words) discussing the results of your experiments (if you have a smaller team, this section will be shorter). Which combinations of improvements outperformed the baseline? Which was the best overall? Which improvement is responsible for the most gain (or loss) in performance? How do they differ, eg. does one improvement drastically improve precision at the cost of recall while another gives small improvements to both precision and recall?

Finally, show 1 or more examples of test input/output pairs from your best model and discuss what strengths or weaknesses of your model are demonstrated by each example. What types of errors does your model make? What does it do well?

4.7 Conclusion – 5 points

1-2 paragraphs (100-200 words) summarizing what you have accomplished in your project. Did any of your models achieve state-of-the-art performance? Looking back, is there anything you would change in your approach, eg. a different improvement or dataset? If you were going to continue working on this task, what is the next approach you would try?