revised: April 21, 2019

Class Project: Interpretability of your NLP Classifier

CSE 256: Statistical NLP: Spring 2019

University of California, San Diego

Due: June 3 - June 7 2019

If your NLP model recommends actions, for example, that the user should invest in a particular stock based on today's news, or that a college administrator should admit a student based on a recommendation letter, it is critical that the user trusts the model. More generally, in high stakes decision making, predictions of machine learning models cannot be acted upon on blind faith as the consequences may be catastrophic.

Interpretability has implications for the adoption of language technologies. In other words, if users do not trust a prediction made by a machine learning model, they are unlikely to use it. In all programming assignments, we evaluate our models using metrics on test data. In this project, we will explore providing explanations for our model's predictions. The goal is to provide some insight into your model's decision making process. For example, if your model is very confident that a particular review is positive, it should help the user understand why your model is so confident about its prediction.

1 Classifier Explanations (40%)

Take the classifier you developed in PA2[1], and add functionality to produce explanations for your model's predictions. Why did your model make the prediction that it did? To generate explanations, it is helpful to think about the features and representations employed by your model. Think about the parameters of your model. It also helps to think about it from a debugging perspective: what would help you understand what is working and what is not working? However, remember that the user might not be a programmer, and your explanations should be as clear as possible.

Try to provide explanations for different cases, for example: when your model is sure and correct; sure but incorrect (overconfident); and when your model is uncertain, etc.

This part should be done on the reviews dataset we provided.

2 Second Classification Task (40%)

Find a dataset for a different text classification task. You are free to pick any text classification task for which you can find labeled data. Examples we talked about in class include: What is the reading level of a piece of text? Will a piece of legislation pass? What is the mental state of the author of the tweet: depressed or happy (could use emojis as labels)? What is the political affiliation of the author? Is this news article (or statement) factual or not? Female or male author?, etc.

Here again your model should provide explanations for its predictions.

3 Creativity (20%)

Be creative in making your explanations and how you display them: comprehensive, clear and intuitive. In this part, you are free to be creative along any of the following dimensions: explanations or visualization or choice of classification task.

For visualization, you can create a simple HTML-based interface that allows making calls to your model. The user provides some input text; your model should make predictions on the input text and show appropriate explanations in the interface. We recommend Django, but you are free to use any tools you are familiar with and prefer.

[1] If you did not get a solution you are happy with, it will be considered adequate to work with the starter code.

4 Prizes

There will be a total of four prizes awarded. The first three prizes will be determined by the vote of the teaching team. The fourth prize will be determined by the class vote.

5 Submission and other Instructions

- **Code:** You will submit your code together with a neatly written README file with instructions on how to run your code.
- **Report:** There will be no formal report, but we will require a few screenshots of your system working, up to 10 pages are allowed.
- **Demo:** We will grade the final project during the lecture time slot on the last week of lectures, week 10, over a three day period. You are expected to show up on all three days to see other demos and to vote. A demo guideline will be posted on Piazza.
- **Deadline:** The demos will take place on June 3 - June 7. The last day to submit your code and screenshots is June 7.
- **Teams:** Teams can have up to three people. Enter your team information in the following Google team by April 26, 2019. `https://forms.gle/B1xgo2sAjVUhzWkE8`