KETTLE.AI BOT RESPONSE AND MESSAGE BUFFER IMPROVEMENT

Adrian Bjune
November 5th, 2018

Objective

The objective of this project is to improve the automation of the Kettle.ai product.

Goals

The end goal of this project is twofold:

1. To identify and emphasize “Call to Actions” sent to users during Deep Work.
2. To build a confident enough model to automate which messages are sent through while a user is in “Deep Work.”

Data

Kettle.ai has graciously allowed me access to their data, which consists of, but not limited to:

- Slack messages to and from Kettle users
- Deep Work start and stop times
- Responses to bot messages

Project Outline

I plan to tackle this project using the following steps:

Part 1:

1. Pull down as much data as I can from the SQL server that I’ll be given access to by Kettle.
2. Determine what features I will use in my model. Though this will take some experimentation, some potential features might include:
    - Total word count
    - TF-IDF Vectors
    - Word Count Vectors
    - Verb Count
3. Use Bootstrap Labelling[1] to create a starter set of labelled messages (1000 < n < 5000) of whether the message is a “Call to Action” or not.

3a. Fit a model to the labelled data. Some models to be considered:

- Naive Bayes
- XGBoost
- Word2Vec
- Convolutional Neural Network

3b. Run the best fit model on the rest of the data, pulling out the highest confidence labels, double checking to make sure that the label is accurate, add those to the training dataset, then repeating the fit and pulling of data until I have a labelled dataset of at least 10,000 points.

4. Once I have a large enough labelled dataset I can do a final GridSearchCV to find the best model to predict whether a message is a “Call to Action” or not.

Part 2:

1. Do EDA on the data I’m able to obtain to find out exactly how much of the data is labelled as “urgent” or “not urgent”

1a. If applicable, combine tables to determine whether a message is “urgent” or “not urgent.”

1b. Examine a subset of the labelled data to determine how accurate the labels are.

1c. If necessary, manually label a subset of the data as “urgent” or “not urgent” for training.

2. Determine what features I will use in my model. This will be similar to the features in Part 1, although I may find that there is signal coming from other features. Some of the other potential features might include:

- Upper Case Word Count
- “Call to Action” Flag
- Punctuation Count

3. Fit and tune a model to predict whether something is urgent or not. The initial models I’ll be testing will be the same as those used in Part 1, although like in Part 1 those are just a guideline for what might work, and I may find another model that does a better job of predicting the target data.

References

[1] Scott E. Reed & Honglak Lee. Training Deep Neural Networks On Noisy Labels With Bootstrapping