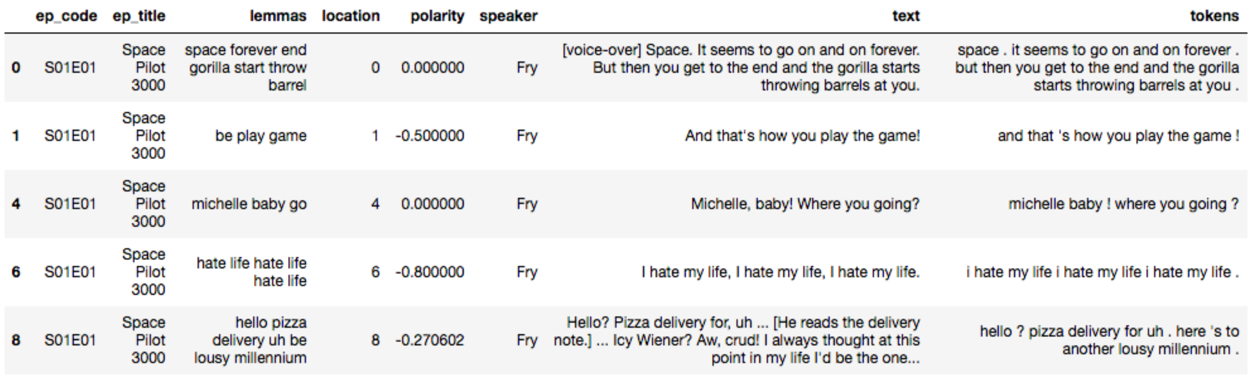
## Project design
The goal of my project was to build a system that could recognize fictional characters using only their words. I focused primarily on the cartoon *Futurama*, though I also ran the algorithm on a another show, *Buffy the Vampire Slayer*, to see if it performed differently on a show with somewhat more complicated characters.

## Tools
I used `sklearn` and `keras` to implement the two learning algorithms, `spacy` for text processing, `bs4` for scraping, and `pandas` for data manipulation. I also used `textblob` to do sentiment analysis, but didn't end up using that data in either of the models I presented.

## Data
I scraped the transcripts of most of the episodes of Futurama from TheInfoSphere.org. One of the movies and a handful of episodes in season 10 have inconsistent page structures and I didn't feel it was worth the time to write a custom scraper for each of them. In total, the scraped episodes contain 24,034 lines of dialogue and 266,538 spoken words.

I scraped the Buffy dataset from BuffyWorld.com. It is complete, as best I can tell, and contains 42,167 lines of dialogue and 386,209 spoken words.

After scraping, I converted the text data to lemmas for the naive Bayes classifier and tokens for the neural network. The resulting `DataFrame` is presented below.



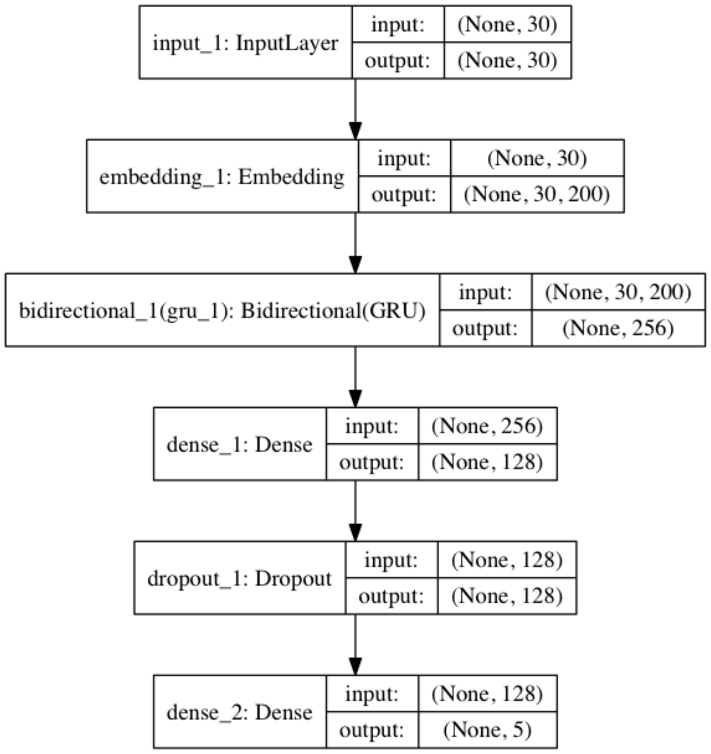
| | ep_code | ep_title | lemmas | location | polarity | speaker | text | tokens |
|---|---|---|---|---|---|---|---|---|
| **0** | S01E01 | Space Pilot 3000 | space forever end gorilla start throw barrel | 0 | 0.000000 | Fry | [voice-over] Space. It seems to go on and on forever. But then you get to the end and the gorilla starts throwing barrels at you. | space . it seems to go on and on forever . but then you get to the end and the gorilla starts throwing barrels at you . |
| **1** | S01E01 | Space Pilot 3000 | be play game | 1 | -0.500000 | Fry | And that's how you play the game! | and that 's how you play the game ! |
| **4** | S01E01 | Space Pilot 3000 | michelle baby go | 4 | 0.000000 | Fry | Michelle, baby! Where you going? | michelle baby ! where you going ? |
| **6** | S01E01 | Space Pilot 3000 | hate life hate life hate life | 6 | -0.800000 | Fry | I hate my life, I hate my life, I hate my life. | i hate my life i hate my life i hate my life . |
| **8** | S01E01 | Space Pilot 3000 | hello pizza delivery uh be lousy millennium | 8 | -0.270602 | Fry | Hello? Pizza delivery for, uh ... [He reads the delivery note.] ... Icy Wiener? Aw, crud! I always thought at this point in my life I'd be the one... | hello ? pizza delivery for uh . here 's to another lousy millennium . |

## Algorithms

I used multinomial naive Bayes as a baseline because it's easy to train and can handle tens of thousands of features.

For my second algorithm, I used a deep learning model using gated recurrent units (GRUs) to convert each line of dialogue into a fixed dimensional thought vector, which I then fed to a standard neural network classifier with one hidden layer. The specifics of the network's structure for the 5 character condition is presented in the graph on the next page.

To speed up training, I used pretrained 200d  GLOVE embeddings I downloaded from nlp.stanford.edu/projects/glove/.



## Results

All metrics are on the test set. Precision, recall, and F1 scores are weighted averages of the per-class scores.

| Naive Bayes Bender vs. Fry | |
|---|---|
| **Accuracy** | 65.46% |
| **Precision** | 65.57% |
| **Recall** | 65.45% |
| **F1** | 65.42% |

| Naive Bayes Futurama main cast | |
|---|---|
| **Accuracy** | 39.94% |
| **Precision** | 41.11% |
| **Recall** | 39.94% |
| **F1** | 40.23% |

| NN model Futurama main cast | |
|---|---|
| **Accuracy** | 41.66% |
| **Precision** | 41.19% |
| **Recall** | 41.66% |
| **F1** | 41.48% |

| NN model Buffy main cast | |
|---|---|
| **Accuracy** | 40.34% |
| **Precision** | 39.86% |
| **Recall** | 40.34% |
| **F1** | 39.20% |

## What didn't work

1. The neural network model didn't work as well as I'd hoped.
2. A significant portion of the dialogue was less separable than I expected. I've seen all the episodes, but when I tested myself I was only able to correctly classify 80 out of 100 examples in the two class Bender vs. Fry condition, and 11 of those I could only identify by recalling events of the show, which the algorithms didn't have direct access to.
3. The sentiment wasn't useful, probably because `textblob`'s builtin sentiment analyzer wasn't (or at least didn't seem) terribly accurate, and both algorithms should already be incorporating sentiment indirectly.
4. I used PCA to reduce the dimensionality of the feature matrix from 65,887 features to 1000 features and then tried using gaussian naive Bayes on the resulting vectors, but this didn't perform as well as using the full feature set.
5. I tried using tf-idf weighting on the term frequency matrix, but that didn't help.
6. I tried using a neural net that used a series of 1D convolutions, but it was slightly less accurate than the sequence based model I presented.