

CS 4701 Final Report

Political Identity Chatbot & Classifier

Team Members:

Aditya Jha (aj377), Leeds Rising (lar294)

AI Keywords

Natural language processing; Neural networks; Deep Learning; Naïve Bayes; Logistic regression; Random decision forests

Application Setting:

Our conversational chatbot can be used in two different settings.

The first is as a conversational, question-answering chatbot to understand different political leanings. The intent is to allow users to pick a political identity for our chatbot (republican, democrat, liberal) and then engage in conversations with the chatbot to understand how individuals with this political identity discuss certain issues.

The second is as a classifier of what political identity the user is. This classifier takes in user text and makes a prediction as to what political identity the user identifies with. The intent is to apply this classification during the user's conversation with the chatbot. This can improve the quality of our chatbot (in terms of its awareness of the user) and potentially be valuable for those who were previously uninterested in politics to understand how their ideologies match with different viewpoints.

Other Students:

For the evaluation portion of our chatbot, we had a number of our peers, across the broad spectrum of different political identities and of diverse backgrounds, serve as user study participants. Their reflections on our chatbot are available near the end of the report.

Note, these are Cornell students but are not in the CS 4701 class:

- | | |
|----------------------------|------------------------------|
| - Adam Kanwal, ask272 | - Yvonne Schichtel, ys397 |
| - Patrick Han, ph367 | - Warren Blood, wmb87 |
| - Cole Williams, crw234 | - Max Graham, mbg224 |
| - Warren Blood, wmb87 | - Diana Finkel, djf259 |
| - Michael Maola, mm2588 | - Niko Cruz-Marsted, nc376 |
| - Blair Legere, bel74 | - Benjamin Haowen Shi, bhs9 |
| - Valentine Starnes, vs344 | - Michelle Abramowitz, mka66 |
| - Lukasz Cegla, lc743 | - Mustafa Hamdani, mah478 |
| - Fem Woodruff, fw223 | - Ellie Month, em669 |
| - Georgia Manning, gcm48 | - Tadiwa Musiwa, tm522 |

Part 1: Project Description

What we planned to do:

Project Proposal Plan

Our initial project proposal at the start of the semester was to build chatbots trained on subreddits with different political leanings. Specifically, we planned on starting with the three of the most popular subreddits in terms of active users and relatively distinct political ideologies - these were the democrat, republican and liberal subreddits. As mentioned in the above application setting portion, our intention was to give users the ability to understand various political viewpoints and if we were successful with the first three most popular subreddits, we wanted to build a system such that the user could enter any political viewpoint of interest and our chatbot could be trained on the corresponding subreddit so the user could then engage in a conversation. The data from our chatbot would come from previously scraped reddit datasets available in Cornell's Convokit library and from our team utilizing the Reddit API.

We planned on building this chatbot with a sequence to sequence model that has previously proven to be effective in chatbot development. The sequence to sequence model utilizes encoder and decoder recurrent neural networks to both take in information and generate output for users. We'll elaborate further on the AI components of the sequence to sequence model in the Key AI Aspects section below.

Once our chatbot had been built, we planned on developing evaluation methods and conducting significant user testing to examine the quality of our system, with a focus on the system's efficiency and satisfaction. Users would fill out a rubric we designed to evaluate the system and based on their feedback we planned to iterate on our model; a further discussion of our system's evaluation is provided in Part 2 of this report.

Additionally, while building this chatbot we had wanted to experiment with how data quality and common optimizations of the sequence to sequence model (teacher forcing, beam search, bucketing) would impact our results. We planned on creating divisions in the data based on reddit comment quality determined by the number of upvotes it got and then using just the highly-upvoted comments as the training data for our model and evaluating our system.

To summarize, at the start of the semester we planned on building a chatbot development system using sequence to sequence models trained on political subreddits that the user could enter. Then, we planned to improve the quality of our chatbot by creating a set of evaluation metrics and conducting user testing while modifying the input data for our chatbot and implementing the most common optimizations of text generation algorithms.

What our project wound up being (Project Plan Update & Final Status):

Project Plan/Status Report Updates

Following the academic pause in coursework, we updated our initial project plan after experiencing very long training times and some difficulties improving our full sequence to sequence chatbot model. We also reduced the scope of a few components of our project to better align with the shift to virtual learning and address the gap in the semester, but added a classifier component following feedback from users of our baseline system.

To begin, we found that the full sequence to sequence chatbot took around ~40 hours for the model to train with our data for one subreddit of 360,000 pairs. This made it difficult to iterate on the model as most of the improvements we had planned to make were related to changing the input data (using comments that had high scores from user votes, excluding conversation threads that had highly negative sentiment or explicit content, etc.). Another aspect that added complexity was that evaluation of the chatbot largely came from interactive testing but the difficulty here was that if the responses to our questions after training a new model were still similar, it was hard to decipher why the model stayed the same due to the black-box nature of chatbot evaluation. Because of this, we eliminated the planned model optimizations and number of iterations on the sequence to sequence chatbot.

In terms of data, we removed the Reddit API scraping portion of our project with the belief that the scraped datasets for Convokit were already extensive enough for large subreddits. This simplified our preprocessing structure by not having to manipulate Reddit API data to match our structure. We also reduced the amount of external user testing on the sequence to sequence chatbot in comparison to our baseline chatbot as our team's testing of the model saw that its answers seemed to lack variety.

In response to user feedback from the baseline model, we decided to build a classifier that would predict the political alignment of the user. This came from comments in a few of our evaluation forms that the chatbot would have greater awareness and emotional affects (two of our evaluation criteria), if it seemed to know more about the user. We thought this was an interesting way to add personality to our chatbot and experiment with other AI models.

Thus, following the instruction in classes, we updated our plan to eliminate model optimizations, reduced the amount of testing, limited data sources to previously scraped data, and decided to experiment with building a classifier.

Final Project Status

Our final project has three separate components - a baseline chatbot that utilizes the Chatterbot library, a deep learning chatbot using a sequence to sequence model and a classifier that takes in text and tries to determine whether the speaker aligns with democrats or republicans. We'll now briefly describe each component of the project, the associated functionality and some general strengths and weaknesses.

Beginning with the baseline chatbot using Chatterbot, we wanted to have a system that could be used as a direct comparison against our final deep learning chatbot. We added a GUI for users to use this system and each Chatterbot is trained on conversations from a specific political subreddits and responds to inputs from the user. Since the training data is only the political subreddit, Chatterbot is a very limited model - it can't respond to non-political questions and its responses are verbatim comments from the training dataset. Chatterbot decides on a response by looking at the input query, searching the training data for the closest match to the input query and then returns the most likely response for the pairs that best match the input query. As a result, the baseline system is effective only if the input query is very similar to a statement in the training data but ineffective if the input query is anything else. Examples of chatterbot usage are provided in the Evaluation section.

Then we moved onto a sequence to sequence model for a more refined chatbot. This chatbot also takes in text query inputs and responds with a text output through the terminal, in addition to a score representing how "accurate" the chatbot's response is relative to our test set. This chatbot is initialized with a general reddit corpus of more than 3 million comments to allow the bot to respond to more generic questions outside the domain of politics, and then trained on specific political subreddits to give it a "political identity". Because of the additional data provided to this sequence to sequence model, it is far better at responding to non-political questions. Its responses are also not verbatim comments and instead seem to be different commonly used words from the political subreddit to form a sentence, with several examples provided in the Evaluation section.

We also built a classifier to predict whether a user's views aligned with either the republican or democratic party. After testing multiple different classifiers described in the evaluation portion, we chose the Naive Bayes classifier with the user's comments and their associated politeness as the feature set. Overall, we thought this was a generally good classifier that impressed our users in a second round of testing as it was correct 8 out of 10 times at predicting our user's political alignment after a short conversation (on average, 6 text inputs from the user). One of the weaknesses of this classifier was an inability to pick up on sarcasm which is further detailed in Classifier Evaluation. Additionally, the classifier's input/output is not linked to the chatbot (requires manual data manipulation in a submitted Jupyter notebook file to get the output).

Key Aspects of AI:

There were two significant aspects of AI core to the project - primarily a sequence to sequence model utilized to build our chatbot and feature creation/selection within our data for various AI classifiers before picking a specific model used to identify whether our speaker aligned with the democratic or republican party.

The sequence to sequence model uses encoder and decoder RNNs to function. Specifically, the input sentence is first fed into the encoder RNN which builds a vector representation that encapsulates the input information. Then the decoder RNN generates a variable length text response to produce an output value. The parameters that allow for this encoding and decoding to take place are tuned during the training of our chatbot, and took several hours (on paperspace) despite the relatively small dataset.

The different classifiers we tested included Logistic Regression, Naive Bayes and Random Forest. Although we did not modify the different classifiers from their standard implementations provided in the Sci-kit learn library, we were primarily curious which classifier would work best for this political classification problem and how changing the provided feature set would make a difference in the results. As we will discuss further in the evaluation portion, we had three different features: the text of a user's comments, the politeness of their comments, and the score their comments received on Reddit which we combined in different ways and evaluated the classifier's final performance on.

Part 2: Evaluation

Our goal was to evaluate our chatbot's performance. However, we quickly learned that throughout history there have been many approaches to evaluating the quality of artificial intelligence systems. After reading a few papers (provided in the resources) on effective chatbot evaluation, we decided to focus on two main components – efficiency and user-satisfaction. Within these two components, we made further subcategories that we will define and discuss below to comprehensively review our system.

Efficiency is a highly regarded assessment when it comes to chatbots. We defined efficiency on the divisions of *Communication*, *Performance*, *Accuracy*, and *Awareness*. Our chatbots' *Communication* skills were based on the chatbots ability to hold up conversations with the users by responding to the users' statements and upholding proper English grammar while doing so. The *Performance* category regarded the chatbots' ability to answer the questions in task and to steer the conversation under the purpose of our project. In addition, we believed we needed to assess the *Accuracy* of the responses our chatbots returned– if the chatbots were replying with real news or if they were making up fake news. We then further assess the *Accuracy* by indicating if the chatbots showed knowledge (a *Awareness* division) of current events and social climates.

User-satisfaction ties in with the Turing Test. A key focus in artificial intelligence is replicating the convenience of human-like interaction. We wished to assess the human-like qualities of our chatbots. Thus, we divided user-satisfaction into the categories of *Consistency*, *Emotional Effects*, and the *Turing Test*. With *Consistency*, we believe that our chatbots had to be able to effectively take nonsensical user-input considerations, understand graceful degradation, and tolerate derailing/manipulation. We believed that our chatbots should be able to sense the users' emotions and feelings towards certain political subjects. We put the chatbots' sense of personality, restfulness towards privacy, and responsiveness to users' mood as qualities for *Emotional Effects*. With the *Turing Test*, we simply asked the evaluators if they believed if the provided chatbots were either artificial intelligence or direct human responses.

We believe that if a chatbot scored highly across all of these different categories, then it would be successful at replicating human-like speech and enable users to understand different political leanings by asking questions on various topics throughout the conversation.

Chatterbot Evaluation:

With our evaluation virtues in mind, we assessed our chatbots with user testing performed by a number of our peers across the broad spectrum of different political identities. We were fortunate enough to be able to work with 20 volunteers who agreed to be evaluators for our project. As most of our subjects were not experienced with programming, we found that Zoom screen sharing was the optimal solution of sharing our chatbots to the users. We simply had our chatbot programs running on our screen during a Zoom screen share session, and asked

the users how they wanted to respond to our chatbot at each step. A few of the subjects who did have experience in programming agreed to follow our directions of setting up our code in their own computers on virtual environments. In addition, the three of us also filled out the evaluation.

Through our convenient and accessible chatbot GUIs, we were able to successfully have users test the chatbots. We asked each participant to fill out the survey as shown in Figure 1.

| Category | Description | Performance Rating | | | | | | | | | | | | Additional Comments | |
|-------------------|---|--------------------|-------|---|---|---|---|---|---|---|---|---|----|---------------------|--|
| Communication | <ul style="list-style-type: none">- conversation was clear- proper grammar- responsive to specific questions | Poor | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | Optimal | Didn't really make sense most of the time |
| Performance | <ul style="list-style-type: none">- ability to answer all questions- complete tasks- steering the conversation- sensible to beliefs and ethics- awareness of social climate | Poor | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | Optimal | Was on subject a few times, but most of the time completely random. It was politics related though. |
| Accuracy | <ul style="list-style-type: none">- providing accurate information- consistency with political affiliation | Poor | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | Optimal | Though it was random a lot of times, it wasn't speaking complete nonsense. It talked about recent politics |
| Awareness | <ul style="list-style-type: none">- sensible to beliefs and ethics- knowledge of social climate and current events | Poor | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | Optimal | It seems to know about social climate and current events |
| Consistency | <ul style="list-style-type: none">- nonsensical user-input considerations- toleration to derailing and manipulation- graceful degradation | Poor | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | Optimal | There was no consistency |
| Emotional Affects | <ul style="list-style-type: none">- sense of personality- sensible and responsive to user's mood and social cues- respectful to privacy | Poor | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | Optimal | Personality was too sporadic |
| Turing Test | was this a human or an artificial intelligent bot? | AI | Human | | | | | | | | | | | | obviously not a human lol |

Figure 1: A sample peer performance evaluation we received back on Chatterbot

The results of Chatterbot can be shown in Figure 2. Chatterbot performed poorly in *Consistency* and *Emotional Effects* receiving average score below 1. Prior to having user testing, we knew that we would slack in these divisions by our observations during the group's self-evaluation. Chatterbot had an average score of around 2 two *Communication* and *Performance*. However, looking into the statistics results, the median is much lower than the mean and there are max scores of 4 for both *Communication* and *Performance*. We believe that Chatterbot was able to perform near "Average" in these categories on the rare occasions. Chatterbot excelled most on *Accuracy* and *Awareness*. Though lacking in Communication, the algorithm we implemented in Chatterbot can nearly constantly bring up accurate current topics. Finally, the *Turing Test* results (though not shown in Figure 2) had that all but one evaluator believed that the chatbot was an artificial intelligent system. It should be noted that we reached out to that evaluator who stated that they only believed the chatbot was a human because of the "level of ridiculousness" the chatbot's responses were.

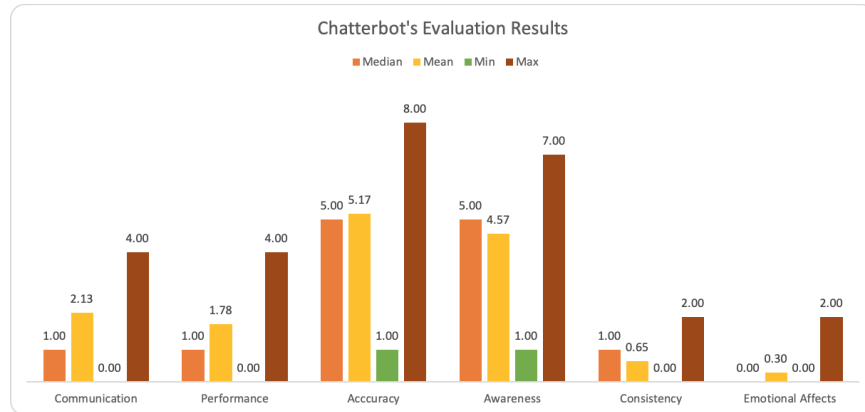


Figure 2: Bar graph of Chatterbot's performance in each category

The algorithm we used for Chatterbot has its pros and cons. Looking through the comments we received on Chatterbot's evaluation, we saw these words repeated often—"random", "weird", and "abrupt". The off-topic responses from Chatterbot attest to these comments. A prime example of a typical response Chatterbot provided is shown in Figure 3B. We see the max scores of 4 for *Communication* and *Performance* because of the rare occasions that Chatterbot is able to return an appropriate response as shown in Figure 3A. Though Chatterbot returned "random" responses frequently, we see that the chatbot excelled in *Accuracy* and *Awareness* because it showed knowledge of current events and social climates. We see in the examples in Figure 3B that even though the responses were unrelated to the question, they were still related to politics and real social issues. Additionally, a lot of the comments stated that there was less randomness with specific and longer questions. Chatterbot performs better with more user inputs as it will be able to use these keywords to find a relatable response. Looking at Figure 3A, we see that Chatterbot was able to return desired responses with a more direct question (even though the occasions of these responses were not so frequent).

You: Can you rate Trump's performance as POTUS in the last year?

MyBot:: He's doing a fantastic job so far in my opinion. Trump is doing exactly what I wanted him to do. I couldn't be happier with President Trump. He may be failing at what you want, but as far as those who supported him and made him president; he's doing great. That's how our government works, just because you don't like it doesn't mean it's a failure. We won, you lost. Suck it up and tell your party and/or representatives to quit acting like lunatics and maybe you'll win some seats back. But the train is off the tracks on the left and the leftist ideology has been rebuked by the American people.

You: Can you rate Trump's performance as POTUS in the last year?

MyBot:: Did you know Trump has the best golf/work ratio out of all presidents so far? Proof that he's doing a great job.

Figure 3A: Successful usage of Chatterbot (rare examples, very direct question necessary)

You: What is your purpose?

MyBot:: Let's look at a possible scenario. The Republican nominee ends up being Rick Perry and Obama is the given Democratic offering. Huntsman decides to run as an independent. Can I take the chance that splitting the left/moderate vote doesn't allow Perry to win? I am no fan of Obama but the idea of a Perry Administration is enough to convince me John Titor is real.

You: Can you rate Trump's performance as POTUS last year?

MyBot:: Private health insurance does not become illegal in most countries with universal coverage. In Germany they have mandated health insurance w/ private providers and they are currently paying 1/2 of Americans. Going further into the free market is reckless, and any plan that is not universal will not be the plan implemented to fix the mess that is the ACA. There are conservative universal plans.

Figure 3B: Unsuccessful usage of Chatterbot (very common example, most inputs to the system)

Sequence to Sequence Model Evaluation:

As we did with Chatterbot, we approached the Sequence to Sequence Model evaluation with the same 20 volunteers and ourselves. As the software and program required to run this model was quite difficult to set up on our users' computers, we used the Zoom screen share approach to allow the evaluators to access this chatbot. The results of the Sequence to Sequence Model evaluation can be shown in Figure 4.

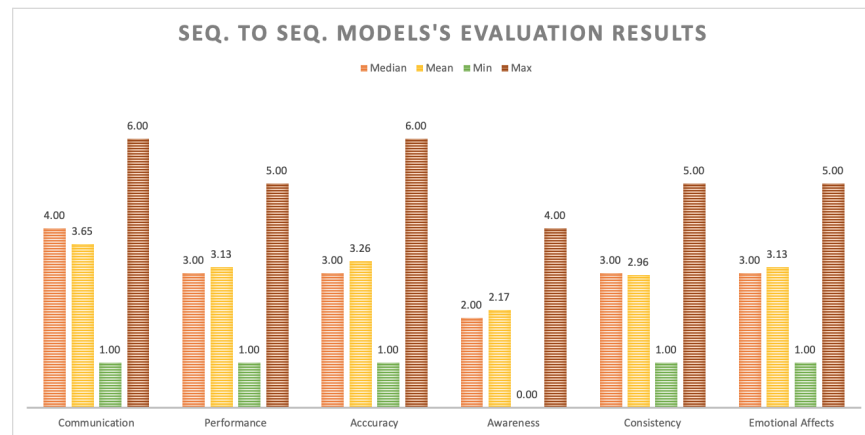


Figure 4: Bar graph of Sequence to Sequence Model's performance in each category

It is apparent that the Sequence to Sequence Model performed far differently than Chatterbot. There are significant improvements in the *Communication*, *Performance*, *Consistency*, and *Emotional Effects* sectors but at the cost of underwhelming in *Accuracy* and *Awareness*. After reading the comments from our evaluators, there is a general consensus that the Sequence to Sequence Model was easier to chat with as it was responsive to direct questions; however, unlike Chatterbot's excellence in maintaining political conversations, the Sequence to Sequence Model failed to keep up political conversations. The screenshotted response shown in Figure 5B plays as an exemplar of this role. Figure 5A displays examples of successful usage of the Sequence to Sequence Model which contributed to the high remarks in the averages scores for *Communication*, *Performance*, *Consistency*, and *Emotional Effects*.

```
> Can you tell me about POTUS?
Trump. [3.42]

> Is President Trump one of the good presidents?
The greatest. [2.37]
```

Figure 5A: Successful usage of the Sequence to Sequence Model

```
> What is your purpose
- I'm not sure that's [-97.65]

> Did Obama do well in his presidential term
- I'm not sure that's the same thing. [4.75]
```

Figure 5B: Unsuccessful usage of the Sequence to Sequence Model

Classifier Evaluation:

Our goal in building a classifier was to correctly predict the political affiliation of an individual using our chatbot. From the CS 4700 discussion on different machine learning models, we tested three different classifiers - logistic regression, naive bayes, and random forest - and attempted to determine which was best for this task of binary political classification.

The data to train the classifier also came from the democrat and republican subreddits. However, this time, instead of looking at pairs of comments and how users replied to each other (the structure used to train our chatbots), we instead compiled all comments a specific user made and then classified the user as republican or democrat. This aligns with the classifier's goal as when an individual uses the chatbot, they will make multiple comments in their conversation but will ultimately have a specific political affiliation.

In testing each of the three classifiers, we modified the amount of data available and the feature sets provided to the classifiers to better understand what led to the best accuracy on the test set. Our original reddit data also included the score of a user's comments (how many upvotes and downvotes they got) which we aggregated into a single score metric. After examining the comment's in the subreddit, we saw that the democratic subreddit had a more civil and generally polite form of conversation when compared to the republican subreddit. We then decided to incorporate politeness as one of the features by running a Convokit transformer over the text data to generate politeness attributes (whether or not a comment showed gratitude, used the word please, etc.) and summed these politeness features into a politeness score feature.

The below table showcases some of the most relevant evaluations of our classifier with the metric of success being the accuracy on a test set (25% randomly selected from total data).

Classifier Evaluation Statistics

Percentages left and right of the dash correspond to the accuracy on the train and test sets respectively

| Feature Set | Data Set Size (# of unique users) | Logistic Regression | Naive Bayes | Random Forest |
|----------------------------|--------------------------------------|------------------------|-------------|------------------|
| User Comments | 2,000 | 99% -- 55% | 93% -- 83% | 87% -- 55% |
| Summed Politeness features | 2,000 | 50% -- 50% | 50% -- 50% | 50% -- 50% |

| | | | | |
|---|--------|------------|-----------------|------------|
| Scores | 2,000 | 53% -- 52% | 49%--49% | 52% -- 52% |
| User Comments, Summed Politeness Features, Scores | 2,000 | 98% -- 56% | 91% -- 57% | 91% -- 53% |
| User Comments | 5,000 | 93% -- 70% | 87% -- 68% | 88%--71% |
| User Comments, Summed Politeness Features, Scores | 5,000 | 93% -- 70% | 87% -- 68% | NA |
| User Comments, Summed Politeness Features, Scores | 10,000 | 86% -- 84% | 93% -- 85% | NA |
| User Comments | 10,000 | 94% --84% | 94% --85% | NA |
| User Comments, Summed Politeness Features, Scores | 17,000 | 91%--90% | 96%--91% | NA |
| User Comments | 17,000 | 91% - 90% | 96%--90% | NA |

Note: NA indicates that runtime was too long to execute locally (greater than 4 hours)

Overall, from our testing, we found that the Naive Bayes model worked best with larger data set sizes and using all of the features. Since the accuracy with and without the additional features are only 1% apart, it is possible this difference is due simply to variance in the test set. However, our intuition is that all of the features helped the model succeed because democrat subreddit users seem to be more polite than their republican counterparts and users of a different political identity do occasionally argue/make points on another political identity's subreddit, but when they do, they receive lots of downvotes leading to negative scores.

That said, we used the Naive Bayes classifier with just comments and politeness with our users because our chatbot had no scoring component. But anecdotally, this was also moderately successful in user testing as even though this was just a binary classification problem and it would not be too difficult for humans to do, it impressed our users. In a second round of user testing with half of our original participants, it correctly predicted 8 out of 10 of the individual's political identity correctly.

From the Zoom calls during this user testing, the cases in which it failed were when users were more adversarial and employed sarcasm - for example in response to a question to evaluate Trump one of user's wrote: No one's like him. He's definitely made America great by not taking action or leading. Our model predicted this user was a republican but was incorrect.

References

Technical Resources:

High level overview of tokenization and sequence to sequence architecture:

<https://medium.com/swlh/end-to-end-chatbot-using-sequence-to-sequence-architecture-e24d137f9c78>

Cohen, D., & Lane, I. (2016, February). An oral exam for measuring a dialog system's capabilities. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (pp. 835-841). AAAI Press. (<https://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/12506>)

Overview of Deep Reinforcement Learning for Chatbots Paper:

<https://arxiv.org/pdf/1908.10331.pdf>

Radziwill, Nicole, and Morgan Benton. "Evaluating Quality of Chatbots and Intelligent Conversational Agents." ArXiv.org, 15 Apr. 2017, arxiv.org/abs/1704.04579.

Classifying Political Orientation on Twitter Paper:

<https://www.aaai.org/ocs/index.php/ICWSM/ICWSM13/paper/viewFile/6128/6347>

Software Resources:

Chatterbot software was utilized to build our baseline system:

<https://chatterbot.readthedocs.io/en/stable/index.html>

YouTube series which we used as a tutorial for our chatbot:

https://www.youtube.com/watch?v=dvOnYLDg8_Y&list=PLQVvvaa0QuDdc2k5dwtDTyT9aCja0on8j&index=1 and corresponding github: <https://github.com/tensorflow/nmt>

Used this resource for example usages of different out of the box AI classifiers:

<https://github.com/jreynolds999/NLP-Reddit-Classification>

Additionally, our application is built utilizing Python. We've leveraged numpy and pandas for data storage and manipulation. Sci-Kit Learn Library is used heavily to create the political classifier model (sk-learn models used are Logistic Regression, Multinomial Bayes, etc.) We also use the Tkinter library to create the GUI.

Data Resources:

Reddit is our primary data source for this project; previously scraped reddit datasets are available: <https://zissou.infosci.cornell.edu/convokit/documentation/subreddit.html>

Generally speaking, we utilize Cornell's Convokit library to manipulate the datasets into our desired format and to get the politeness features: <https://convokit.cornell.edu/documentation/>

Relationship to other work by team members:

No team member is working on a similar project.

Presentation Video:

<https://www.youtube.com/watch?v=Holr1bAZLIIs>