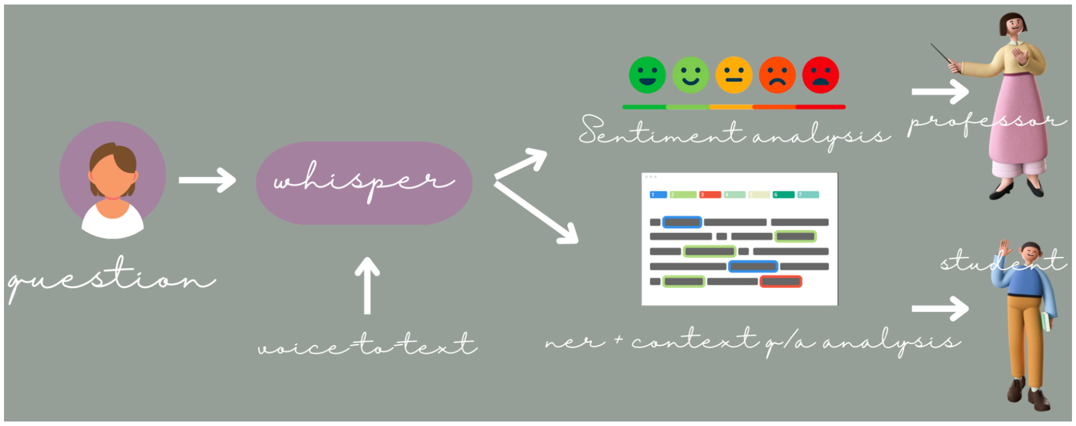
NLP Project Report

Group members: Arnav Joshi and [redacted for privacy]

What did you do & why is it interesting?
We worked on creating a dashboard and a pipeline that uses natural language processing to make question answering analytics easier. We created a platform that takes in audio input of the question, then gets its context and provides a preliminary, supportive answer (using question answering analysis), and named entity recognition (NER). Along with this, we get the sentiment analysis for the professor. It can be depicted as follows:



It is very interesting because it is highly applicable and actually usable in classrooms to assist with learning and teacher support.

What is your data?
We used the following dataset for our models:

- ***Question answering (BERT)***
    - Squad dataset
    - Size: 100,000 questions and their corresponding answers, along with the relevant paragraph of text from which the answer can be extracted
    - Created by researchers at Stanford University from Wikipedia articles and crowdsourcing questions and answers
    - In terms of labeling, each question-answer pair in the SQuAD dataset is labeled with the text passage from which the answer can be extracted. With short, factual answers for the data
- ***Twitter dataset***
    - From NLTK tweet sentiment
    - Dataset is relatively large, with 1.6 million tweets
    - The Twitter NLTK dataset was created by researchers at the University of Copenhagen and Twitter's streaming API

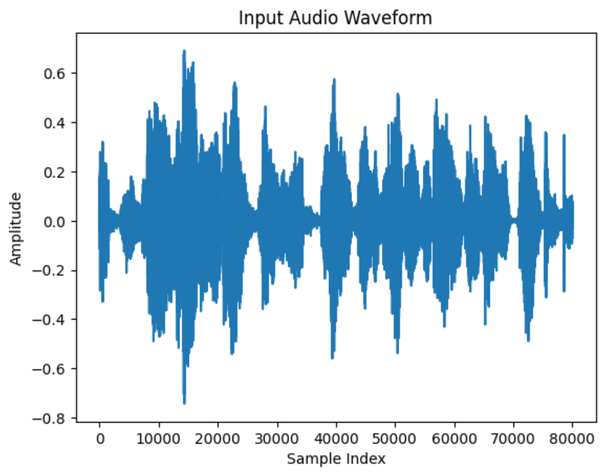
    - They are classified based on a three-way classification of sentiment: positive, negative, or neutral
    - Used with the models:
        - Sentiment analysis: BERT (not-fine tuned)
- ***IMDB dataset***
    - From the HuggingFace datasets library
    - Size: 25,000 positively labeled reviews, 25,000 negatively labeled reviews
    - Created by Stanford researchers by pulling data from IMDB movie reviews. This dataset is substantially larger than the last benchmark dataset produced by these researchers
    - Used with the models:
        - Sentiment analysis: BERT (fine-tuned)
        - Sentiment analysis: Naive Bayes Classifier
- **Annotated Corpus for NER dataset**
    - From kaggle
    - Size: 1.3 million words
    - Dataset to be used specifically for NER from kaggle. Contains an uneven distribution of tagged words
    - Structured as sentences with words that each have a NER label
    - Used with the models:
        - NER: LSTM with Character Embeddings
        - NER: spaCy (for testing)

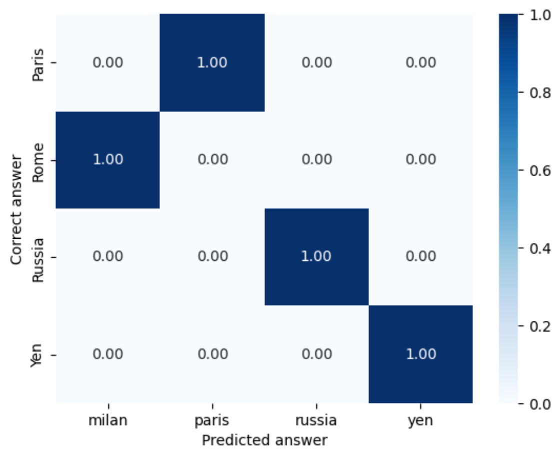
How do your models work?

- ***Whisper Models***
    - We used the whisper model to load in audio input data and transcribe the audio
        - Using Wav2Vec2Processor and Librosa to get the audio input
        - Additionally, using pre-recorded audio files as the input
    - The whisper model is used with a pre-trained model
        - We used a tiny, base, small and medium models that are pre-trained to get the output from the models
    - We finally compared all the outputs to understand the difference in outputs
    - Since the whisper model is a pre-trained model, we did not need to worry too much about the decoding and training process.
- ***Question Answering Analysis Model***
    - We used the pre-trained BERT model on question answering using the SqUAd dataset
    - We used the pre-trained BERT tokenizer to tokenize the inputs to create the actual model
    - We then computed the answer by extracting the answer from model output

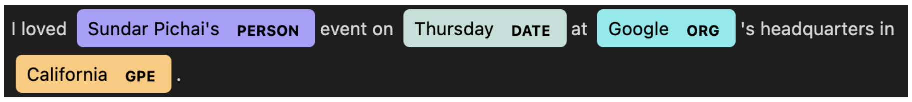
- **Named Entity Recognition (NER) Models**
    - LSTM with Character Embeddings
        - We used the NER dataset to train the model
        - The inputs were padded and character embeddings and word embeddings were created (and reflected in model architecture)
        - The goal of the character embeddings implementation was to give a higher accuracy to unknown words as portions of the word could be used to improve the model's performance (through inferring its meaning from some of its characters)
    - spaCy NER
        - We used the NER dataset to test this model
        - This model is a state-of-the-art pretrained model and has been trained substantially compared to the training we performed on the LSTM with Character Embeddings
        - A pipeline is loaded in and then text can be simply passed to analyze NER
        - We used this model in our final pipeline
- **Sentiment Analysis Models**
    - Naive Bayes Classifier
        - We used the IMDB dataset to train this model
        - This implementation was based on the code for Homework 3 but attempted to be improved using some state-of-the-art tools
        - spaCy lemmatization was a major change which improved performance
        - This model still fell behind most other models in performance
    - Finetuned BERT
        - We used the IMDB dataset to train this model
        - A HuggingFace model (distilbert-base-uncased-finetuned-sst-2-english) was used as the base for this model
        - The model was trained further on the IMDB dataset and performed really well on the dataset, but became very specialized
        - We steered away from using this model despite its performance as we had not settled on a dataset for the pipeline's specific use case, so fine tuning BERT seemed counterproductive
    - Pretrained BERT
        - We used the twitter dataset to test this model
        - We used a pipeline with the base model from the finetuned BERT model (distilbert-base-uncased-finetuned-sst-2-english)
        - This model performed relatively well
        - Great for general purpose usage - we used this model in our final pipeline

What are your results/conclusions?

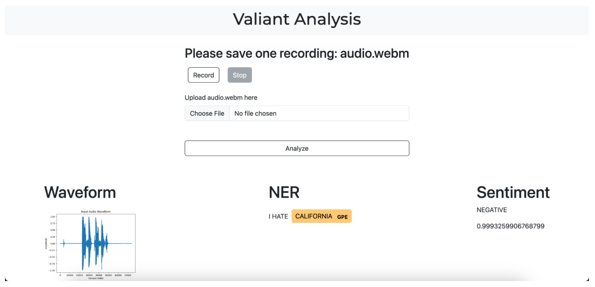
- ***Whisper models***
    - We looked at the different outputs from the different models to understand the transcription performance of the whisper model.
    - We were also able to visualize the voice transcription to look at the voice modulation and the voice effects on the model
        - 

    - We finally, also looked at the predicted and actual text and tried to find the difference in the transcription
- ***Question Answering analysis model***
    - We looked at the confusion matrix based on classification to get an understanding of the expected and true answers, since we used a pre-trained model, we got an expected accuracy of 100% on the basic dataset with the context.
    - 

Correct answer \ Predicted answer	milan	paris	russia	yen
Paris	0.00	1.00	0.00	0.00
Rome	1.00	0.00	0.00	0.00
Russia	0.00	0.00	1.00	0.00
Yen	0.00	0.00	0.00	1.00

- **NER Models**
    - As expected, the spaCy model completely outperformed the model we trained.
    - Most modern NER relies on spaCy as many people cannot create models that outperform it, even finetuning BERT usually performs worse.
    - 

I loved Sundar Pichai's PERSON event on Thursday DATE at Google ORG 's headquarters in California GPE .

- **Sentiment Analysis Models**
  - BERT models tend to perform the best on this task, but this task is incredibly corpus specific which makes it difficult to train a generalized model
  - *[{'label': 'POSITIVE', 'score': 0.9940869808197021}]*
- **Final Dashboard**
  - 
    Valiant Analysis

    Please save one recording: audio.webm

    Upload audio.webm here

    No file chosen

    Waveform

    NER

    I HATE CALIFORNIA GPE

    Sentiment

    NEGATIVE

    0.9993259906768799

  - Performs q/a with NER and sentiment analysis

Future experiments/explorations/additions?

There are a lot of future directions we can take with the project including the following:

- Using the data for the entire class (with the class recordings) and not just the questions. This can help with us being able to figure out which parts of the recording were questions.
- Detecting speaker changes.
- Analyze voice effects and incorporate it into the sentiment analysis.

"Let's take an idiom, such as "feeling under the weather", which is commonly used when someone is ill, to aid us in the explanation of RNNs. In order for the idiom to make sense, it needs to be expressed in that specific order."

"How do we take idioms into account from different cultures?"

"As a result, recurrent networks need to account for the position of each word in the idiom and they use that information to predict the next word in the sequence." [1]

Works Cited

- https://huggingface.co/docs/transformers/main_classes/pipelines
- https://www.depends-on-the-definition.com/
- https://huggingface.co/distilbert-base-uncased-finetuned-sst-2-english
- [1] https://www.ibm.com/topics/recurrent-neural-networks
- https://huggingface.co/blog/sentiment-analysis-python
- https://github.com/openai/whisper
- Squad dataset [https://huggingface.co/datasets/squad]