Disagreement Detection in Spotify Podcast Episodes

Connor Capitolo
Harvard University
connorcapitolo
@g.harvard.edu

Max Li
Harvard University
manli
@g.harvard.edu

Morris Reeves
Harvard University
morrisreeves
@g.harvard.edu

Jiahui Tang
Harvard University
jiahuitang
@g.harvard.edu

Abstract

We study the detection of disagreement in Spotify podcast episodes based on audio data and automatic transcriptions. We apply unsupervised and supervised approaches for disagreement detection to evaluate whether combining prosodic and lexical features can improve detection over utilizing each individually. We also explore which avenues of feature engineering for audio data may be promising for disagreement detection and assess the utility of automatic transcripts. We aim to contribute to the existing literature by utilizing higher dimensional audio features such as the wavelet representation and augmentation, as well as combining features from both the audio and transcripts for disagreement detection.

1 Introduction

With the rise of social media platforms and online forums, the public sharing of opinions to wide audiences is easier than ever; disagreement is both ubiquitous yet also nebulous to strictly define. Disagreement can be instructive both at a societal level, where certain topics can reflect pressing cultural issues, as well as at a dialogue level, where disagreement can reflect the stances of individual speakers. Our goal is to generate insights about disagreement based on the Spotify English-Language 100K Podcast Dataset (Clifton et al., 2020b) using its audio files, transcriptions created by Google's Speech-To-Text API, and corresponding metadata. Disagreement detection is highly beneficial to streaming services as it can lead to higher retention rates and active engagement by its listeners through more tailored recommendations, while also increasing audience traffic and creating a more loyal fan base for podcast creators. This can lead to a better user and creator experience, which may allow streaming services to both expand their platform and consolidate their base. Finally, given the increasing polarization around hot-button topics such as abortion, gun rights, and election integrity, disagreement detection could be used to help streaming services root out inflammatory podcasts that may violate the company's Terms of Use.

2 Related Works

2.1 Domains and media

Previous work on disagreement detection has focused primarily on modeling text data using dependency relations and lexical features such as n-grams. (Xu et al., 2019) utilize pairs of tweets with stance labels (Favour/Against/None) grouped by topic from the SemEval-2016 dataset, defining disagreement as pairs whose stance labels differ. In the realm of online forums, (Gokcen and de Marneffe, 2015) utilize quote-response pairs with agreement scores between -5 and 5 from the Internet Argument Corpus (IAC). Similarly, (Wang and Cardie, 2014) use the Authority and Alignment in Wikipedia Discussions (AAWD) dataset consisting of annotations of disagreement/agreement/neutral on English Wikipedia Talk pages.

Existing works which employ audio data for disagreement detection utilize low-dimensional prosodic features such as speech rate and fundamental frequency. For example, (Hillard et al., 2003) manually annotate transcribed ICSI meeting recordings to create labels such as positive, backchannel, negative. They also use statistics on pause duration, speaker-normalized vowel duration, and fundamental frequency that are employed alongside word-based features such as negative and positive keyword counts to make predictions. Similarly, (Wang et al., 2011) analyze English broadcast conversation data from the DARPA GALE program annotated for 11 (dis)agreement-related labels, utilizing prosodic features including pause, duration, speech rate, and pitch alongside lexical features.

In this context, the Spotify Podcasts dataset stands out in both the volume and diversity of its audio content across various genres and podcast creators, and presents an opportunity to utilize both prosodic and lexical features to explore characteristics of human speech associated with disagreement. Additionally, (Wang et al., 2011) find sarcasm to be a common source of error in disagreement classification; the large volume of combined text and audio data in the Spotify Podcasts dataset may enable progress in this domain as well.

2.2 Definitions of disagreement

A central challenge to disagreement annotation is its inherent subjectivity: while some works in the literature provide explicit definitions of disagreement, most implicitly define disagreement through data processing decisions. (Wang et al., 2011) provide an explicit definition of (dis)agreement: it "occurs when a responding speaker agrees with, accepts, or disagrees with or rejects, a statement or proposition by a first speaker." To build on this definition, we define disagreement as *a speaker directly applying a contradiction or rejection of another person's idea(s) where it is immediately perceptible by the listener*. The rule of thumb is that if the disagreement clip was handed to your mother without any prior context, she would immediately recognize it as disagreement. This allows for handling situations such as sighs and laughter which might otherwise be construed as the onset of disagreement upon rewinding; it also notably (and deliberately) excludes single-speaker podcast episodes, audio segments in which speakers explain the reasons for their disagreement, and audio segments in which speakers respond to soundbites. Our definition hopes to identify segments of disagreement accurately and with a high level of consensus.

Regarding the granularity of disagreement windows, (Wang et al., 2011) and (Wang and Cardie, 2014) define disagreement at the utterance (subturn) level, whereas (Hillard et al., 2003) annotate disagreement at the "spurt" level (period of speech by a single speaker, with no pauses longer than 0.5 seconds). Our definition involves granularity at the utterance level in order to allow for detection of sub-windows of disagreement (towards another speaker within the podcast) within long uninterrupted speech segments, which is common in podcasts. Given the dialogic nature of most podcast episodes, this naturally diverges from that of (Xu et al., 2019) in which disagreement is non-dialogic (independent, differing, stance-bearing tweets on a topic).

2.3 Evaluation Metric

Please note that the F1 score will be reported for this milestone, but subsequent milestones will report the F0.5 score

The F-beta metric was found to be a common evaluation metric across different works, particularly the F1 score; it allows for balancing the weight placed on the precision and recall in detecting disagreement. For the particular task at hand of disagreement detection on podcast data, it is important for streaming services that all podcast search results meet the user's specifications. For example, if a user wants to find a podcast involving lively debate on the Silk Road, it's vital that all podcasts returned from the search include both debate and discussion about the Silk Road; therefore, a more appropriate metric for this problem is the F0.5 score since false positives are more important than false negatives, and a higher priority should be placed on precision than recall. The F0.5 score will be used for comparing across different models, but due to the loss of information from condensing information into a single value, precision and recall metrics will always be reported as well.

3 Data and Preprocessing

3.1 Metadata

The Spotify Podcast Dataset consists of $105,360$ episodes across $18,360$ shows, amounting to around $50,000$ hours of audio and over 600 million words. The podcasts are sampled from Jan 1, 2019 to March 1 2020, filtered specifically for the English language. Since there is a mix of both professional and amateur creators, the podcasts are diverse in terms of audio quality, topics, and structural formats.

We explored the three different types of data (audio, transcript, metadata) to understand their features and distributions. First, the automatic transcriptions data contain speaker tags (based on automatic diarization) and timestamps for each word in the transcripts of the podcasts. It is worth noting that the speaker tags are automatically generated by Google's Speech-To-Text API, which can be inaccurate especially when multiple speakers are talking simultaneously. Second,

the podcast metadata contain some useful features such as episode_title, author, category, subcategory, show_name, show_description, publisher, language, episode_name, episode_description, duration. Note that the metadata is provided by the creators of the shows, so some features might not be very reliable. For instance, creators may incorrectly label their podcasts as a more popular genre to gain a larger audience. The audio from the podcasts will be explored in depth in future sections.

We performed some exploratory data analysis on the metadata of the podcasts, focusing on some of the features mentioned above. In Figure 2, we show a Pareto chart of the total number of shows for each of the 23 major categories found from the metadata. We can see that among the 18,360 shows, Comedy and Education are the two largest categories, while Sports and Health & Fitness ranked third and fourth, respectively. In terms of detecting disagreement, we expect Comedy and Sports podcasts with multiple speakers to contain many disagreement segments; it remains to be seen if the abundance of shows in these categories would yield an abundance of disagreements in the podcasts.

We also noticed that some categories with the least amount of shows have similar names as the top ones, such as Sports & Recreation and Health. As mentioned before, creators may label their podcasts as a more popular genre to get a larger audience. It is possible that the first group of creators preferred one tag over a similar one (for instance, choosing Sports over Sports & Recreation), and then subsequent creators followed these increasingly popular ones.

Looking at Figure 3, we take the same approach as the data description paper (Clifton et al., 2020a) to visualize the episodes that are less than 90 minutes long (there are 125 episodes that exceeded 90 minutes in length that are not shown). We see that most episodes are less than 20 minutes long; in general, the number of episodes decreases as the length increases. Also, a large number of podcast episodes last less than 5 minutes; many of these are trailers or music remixes, in which we would not expect to find a high concentration of disagreements. This plot serves to show our data aligns with the paper description.

As discussed at the start of this section, we would like to use metadata as a guide towards finding podcasts with disagreements so that we can find specific ones for annotation. Our first attempt was to search for the word "disagreement" in the podcast descriptions and listen to those. While this is not the most efficient approach, it was effective in identifying podcasts that contain disagreement.

3.2 Text Transcriptions

3.2.1 Preprocessing

The JSON corresponding to the transcription of each podcast episode was parsed into a sequence of text blocks and a sequence of associated start and end timestamps. Each text block comprises up to 30 seconds of transcribed speech, automatically segmented by Google's Speech-to-Text API based on speaker pauses. For validation of overall data quality, the total transcribed duration vs. word count was plotted as in Figure 4; unsurprisingly, there is a strong linear relationship, though the range of word counts increases up to approximately the 90 minute mark.

3.2.2 Transcript Coverage

As we explored and annotated specific episodes in the data, it was found that the transcript data has coverage issues. Specifically, Google's Speech-To-text API failed to comprehensively transcribe every spoken word. Since transcript coverage can affect the quality of the text data if a large portion of the missed transcript overlapped with the disagreements in the podcasts, it's important to explore where and how often did the transcription fail. First, we approximated the proportion of missed transcripts in the whole text dataset by calculating the ratio between the total duration of the transcribed text and the total duration of the podcast for each episode. We found that the median of transcribed episode proportion was 95.6%. The distribution of the transcribed episode proportions was plotted in Figure 5.

Notice that we only visualized the episodes with calculated transcribed proportion no larger than 1, since those with proportion larger than 1 could be results from inaccurate podcast duration records from the metadata provided by the producers. We do see from the plot that most of the podcasts have a high transcribed proportion, which indicates that our text data has good coverage quality.

We then looked into a specific clip of a podcast and located the missing transcripts to examine if those segments contained disagreements; specifically, we analyzed a 5-minute podcast clip which contained many segments of disagreement between a man and woman on the topic of dog ownership.

We annotated it by labeling the segments of audio data based on our definition of disagreement. Then, we compared the audio with its corresponding transcript. We recorded each word that was not transcribed in the text data with its time stamps and speaker tag; the full list can be found in Table 3.

Since the speaker tags of the missed scripts are less relevant to our discussion in this section, we omitted them for better formatting. In terms of duration, 5.36% of the total excerpt duration was not transcribed; this matched with our approximation of the missed proportion of the whole dataset. We also noticed that most of the missed scripts occurred when the two speakers were talking simultaneously, and words from the speaker with a lower amplitude tended to be missed. This implies that the transcript data could miss some parts where disagreements occurred. To examine this point, we compared the occurrence of the missed transcript and our disagreement annotation. We visualized our comparison in Figure 6.

From this visualization, we can see that there exists some overlap between the missed scripts and our annotated disagreements. For example, the first missed script in the clip contained the line: "it's. It's not". All of us labeled this line as a disagreement in the audio data. Another example occurred at 43:50 where the word "no" did not get transcribed even though three of us labeled it as disagreement.

Hence, we concluded that the imperfect coverage of the transcripts could affect the comprehensiveness and accuracy of the disagreement detection based only on the text data. Therefore, we find it necessary to analyze the audio data and potentially combine the features from both sides for better coverage and accuracy.

3.3 Audio

3.3.1 Audio Encoding

We used the Python package Librosa to explore what kinds of information we could obtain from audio data. Unlike other data sets that naturally come with features, we needed to explore what level and granularity of information we could construct from each audio data file and how to pre-process raw data for model building. Librosa allows for transforming .ogg audio files into waveform metrics and sample rates.

The following analyses are based on a single selected annotated episode which contains many segments of disagreement between a man and woman on the topic of dog ownership, which is a particular 10 minute segment of the entire podcast episode since it contains high levels of disagreement that meet our definition. Three different types of graph (waveform, spectrogram, chromaplot) were visualized to compare whether there is any significant difference between regions of disagreement and non-disagreement.

In the waveform plot in Figure 7, we have time series amplitude metrics with an envelope view in monophonic, stereo, harmonic and percussive decomposition views. There is no obvious visual difference when comparing disagreement with non-disagreement sub-segments of the episode.

We also encoded audio information from signals into time-frequency (Hertz) by plotting a spectrogram, with the x axis being the time series, and y being the frequency measurement of the audio, as shown in Figure 8

For pitch class information, we plotted a Chromagram for the episode in Figure 9, which gives us information about pitch class profile and pitch intensities. It could be particularly useful for analyzing audio whose pitch could be meaningfully categorized.

3.3.2 YAMNet Embedding

We also explored the YAMNet embeddings for the same podcast episode to see if there is any association between our manual disagreement annotations and these vector representations of audio segments. YAMNet is a deep neural network that employs the MobileNetV1 and predicts 521 audio event classes (trained on the AudioSet-YouTube corpus). We utilized pretrained embeddings available through Spotify; there is an embedding of dimension 1024 for each 0.48 seconds chunk. These chunks are visualized via UMAP in Figure 10 together with our annotations. There is no obvious separability based on the scatterplot when visualized in two dimensions, which we hypothesize is due to: (1) 0.48 seconds being too short a window to meaningfully expect differences in audio characteristics, (2) only a subset of YAMNet event classes may be relevant for disagreement, and even then only in rare cases (e.g. shouting would be expected to be limited to only the most extreme of disagreements), and (3) there may be higher dimensional structure to the data that is not discernible from the figure.

3.3.3 Wavelet Transformation

The Fourier Transform is a transformation used to extract frequencies from a signal using a series of waves with different frequencies to analyze a signal. However it has a big drawback - a narrow window will localize the signal in time, but there will be significant uncertainty in frequency; if the window is wide enough, then the time uncertainty increases.

The Wavelet Transform (WT) is another technique for analyzing signals. It was developed as an alternative to the short time Fourier Transform (STFT) to overcome problems related to its frequency and time resolution properties. A Wavelet is a wave-like oscillation that is localized in time. Wavelets have 2 fundamental properties: scale and location. Scale defines how stretched or squashed a wavelet is, while location is a position in time. The advantage that wavelet transforms has over FFT is that it can capture spectral and temporal information simultaneously. A signal is transformed into a set of wavelets at different scales and positions (George Tzanetakis, 2001).

Wavelet transformations have two forms: discrete wavelet transform and continuous wavelet transform. The Discrete Wavelet Transform (DWT) is a special case of the WT that provides a compact representation of a signal in time and frequency that can be computed efficiently. The DWT is only discrete in the scale and translation domain, not in the time-domain. More specifically, unlike the STFT that provides uniform time resolution for all frequencies, the DWT provides high time resolution and low frequency resolution for high frequencies and high frequency resolution and low time resolution for low frequencies. In that respect it is similar to the human ear which exhibits similar time-frequency resolution characteristics (George Tzanetakis, 2001). Continuous wavelet transform (CWT) is a formal (i.e., non-numerical) tool that provides an over-complete representation of a signal by letting the translation and scale parameter of the wavelets vary continuously.

In this project, we processed our audio data with DWT technique for feature extraction and feed transformed wavelet into our modeling process.

3.3.4 Audio Augmentation

Audio data augmentation is a common strategy adopted to increase the quantity of audio training data, avoid overfitting and improve robustness of the models. In our case, it could also potentially overcome the data scarcity problem. Data augmentation on audio could include Time Stretching, Pitch Shifting, Dynamic Range Compression, and Background Noise (Justin Salamon, 2016).

We used python library Audiomentations for audio data augmentation. It provides functionalities such as *AddGaussianNoise, TimeStretch, PitchShift and Shift* etc.

4 Models

4.1 Text Transcript Modeling

Text transcripts can provide additional features beyond those directly available in the audio transcripts since they encode semantics and lexical features such as phrases associated with disagreement.

Count-based baseline model

Following the approach in the literature, we utilize as our baseline a count-based model, hypothesizing that higher counts of negation words in a transcript segment corresponds to higher probability of disagreement in that segment. Four categories of negation words are considered: analytic negation ('no', 'not'), contraction negation (e.g. 'didn't', 'shouldn't'), synthetic negation (e.g. 'neither', 'never', 'nobody'), and other words which we hypothesize to be associated with disagreement ('disagree', 'incorrect', 'wrong', 'ridiculous', 'absurd').

The distribution of these words across all 8.4 million transcribed text segments is displayed in Figure 11. We observe that usage of these negation words is relatively rare, and that analytic negation ('no', 'not') is used more frequently than other negation categories. Based on these observations, we consider two count-based models: one which only utilizes analytic negation counts (which we refer to as the 'lexicon reduced' model), and one which utilizes the total count across all negation categories (the 'lexicon full' model).

Based on the distribution, we classify a given text segment as 'disagreement' if the count of negation words is greater than 1 (for the lexicon reduced model) and greater than 2 (for the lexicon full model).

For evaluation, we compute the union of annotations on a single test episode (with id 79aiOALBohH9vEIUHPAoRj, discussing dating and food in NYC). Annotations are joined with the text transcript segments based on overlap: the true label of 1 is assigned to a text transcript segment if there

is any overlap between the text transcript and annotation timestamps. Table 1 details the results.

Model	Precision	Recall	F1
Lexicon reduced	0.27	0.29	0.28
Lexicon full	0.33	0.14	0.20
Cosine similarity	0.24	0.36	0.29

Table 1: Evaluation on test (dating and food) episode

Embedding cosine similarity model

Rather than using a fixed set of words as with the count-based model, we may hypothesize that transcript segments that are similar to expressions of disagreement in some dense embedding space are more likely to contain disagreement. This forms a natural motivation for converting each transcript segment to a vector, and classifying as disagreement based on cosine similarity to a user-defined query (we utilize "No I disagree"): we refer to this as the 'cosine similarity' model.

To convert each text transcript segment to a vector representation, we utilize the all-MiniLM-L6-v2 version of SBERT (Reimers and Gurevych, 2019), a pretrained transformer model which maps sentences paragraphs to a 384 dimensional dense vector space. The pretrained model comes tuned on a dataset of 1 billion sentence pairs (from sources such as Reddit, Stack Exchange, and Yahoo Answers) with a contrastive learning objective: predicting which of a set of randomly sampled sentences was paired with the given sentence in the dataset. By default, SBERT truncates to 128 word pieces; to convert longer text transcript segements to vectors, we take the weighted average of subsegment embeddings.

For classification, we classify a given text segment as 'disagreement' if the cosine similarity between its embedding and the embedding of 'No I disagree' exceeds 0.2. We choose this generic phrase as a query to try to isolate the expression of disagreement from any particular topic. Table 1 displays the results. Although the cosine similarity model achieves higher F1 score than the lexicon reduced and full models, we observe that this is due to higher recall, which partly reflects the low cosine similarity threshold chosen.

In Figure 12, we visualize the 2D UMAP projection of the SBERT embeddings. We hypothesize that one reason for the modest performance of the cosine similarity model is that the embeddings may capture similarity of general areas of discussion rather than speakers' stances towards the areas of discussion. For example, the bottom right cloud of red points (corresponding to annotated disagreement segments) corresponds to discussion about dogs, and therefore embedding similarity may be capturing this common topic rather than a high incidence of disagreement about the topic.

Going forward, we hope to improve this analysis by exploring ways to separate speaker stance from the topic of discussion, such as by combining the text and audio embeddings.

4.2 Audio Baseline Model

For baseline model in audio analysis, we used a logistic regression model for binary classification into labels of disagreement and non-disagreement.

For input features of the baseline model, we used original waveform, augmented waveform, variance of aggregated original waveform, approximate and detailed coefficient from DWT, each recorded at different timestamp with an interval of 1 second. The training data we used for building baseline model is the annotated dog episode. And the output label of baseline model is classification of disagreement or non-disagreement.

Below is a graph of model performance and its ROC curve, as we could see, it is barely better than a random classifier.

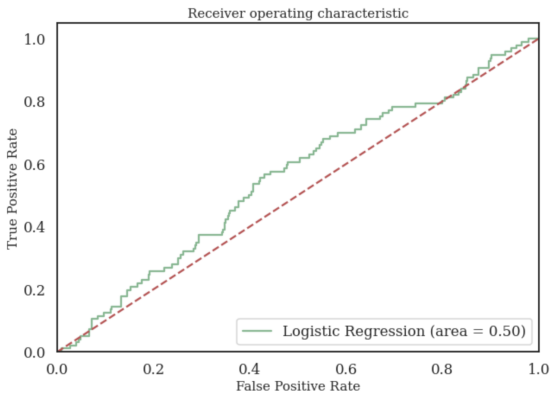

Figure 1: ROC Curve of logistic regression model

A further look into classification report below in table 2 shows that the model achieves the high F-1 score and accuracy by simply classifying all points as non-disagreement. None of the points are being labeled as disagreement. It means the model failed to extract meaningful information from disagreement data points.

In reflection, there is loss of information in the

Label	Precision	Recall	F1
Agreement	0.82	1.00	0.90
Disagreement	0.00	0.00	0.00

Table 2: Logistic regression evaluation on test (dog) episode

way we processed and aggregated data into static points with interval of 1 seconds. In the future, we plan to apply DWT on raw data with higher resolution for each time chunk of 1 seconds. Besides, more data points and annotations are needed. We also plan to use advanced models (i.e. LSTM, RNN) in the next step and incorporate time lag feature in time series data analysis.

5 Appendix

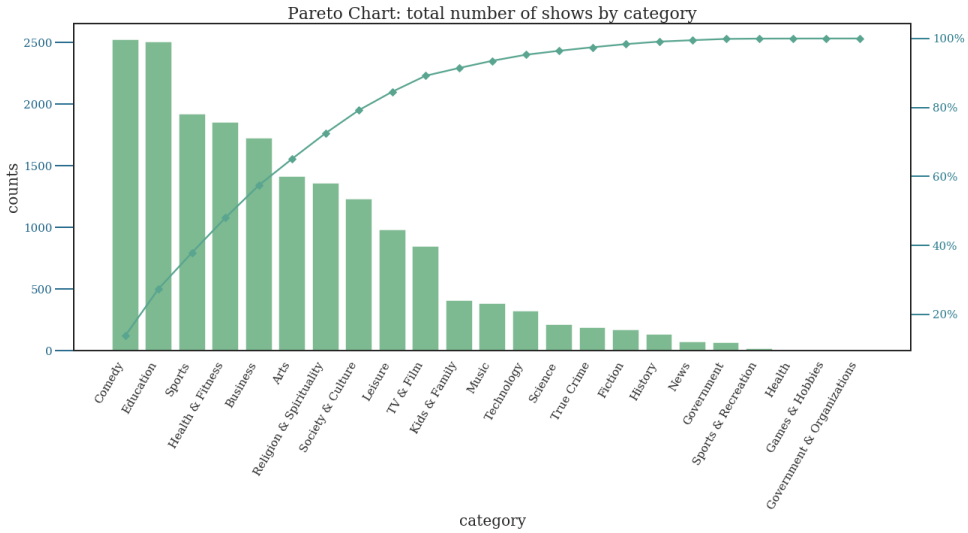

Figure 2: Pareto chart of total number of shows by category

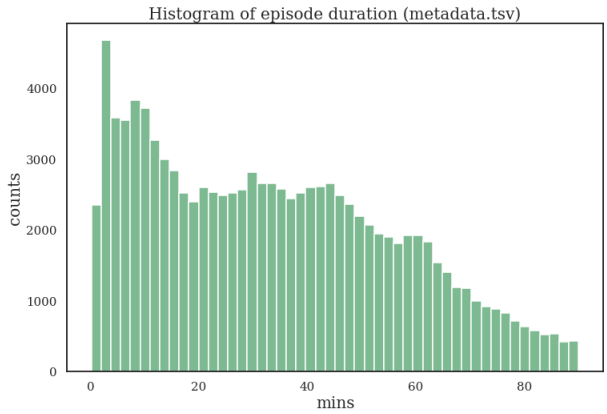

Figure 3: Histogram of episode duration (only episodes shorter than 90 minutes are included for better visualization)

Timestamp	Duration	Utterance
41:10-41:13	3s	It is. It is not
41:33-41:35	2s	Ah. Yes, they are.
42:11-42:13	2s	I understand.
42:21-42:22	1s	Definitely not
43:38-43:39	1s	It's okay.
43:45-43:47	2s	It's not right because
43:50-43:51	1s	No.
44:10-44:12	2s	Wow, that's uncalled for.

Table 3: Utterances missed in transcripts for dog-related clip example

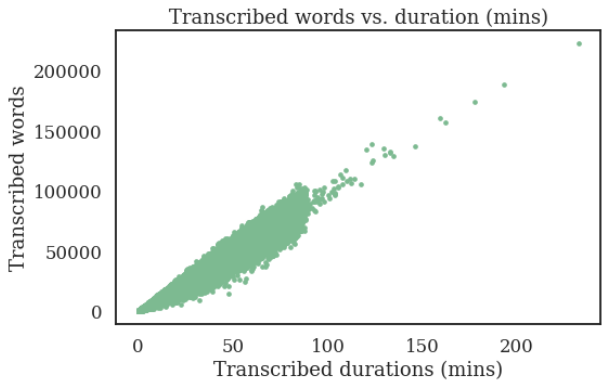

Figure 4: Episode transcribed duration vs. word count

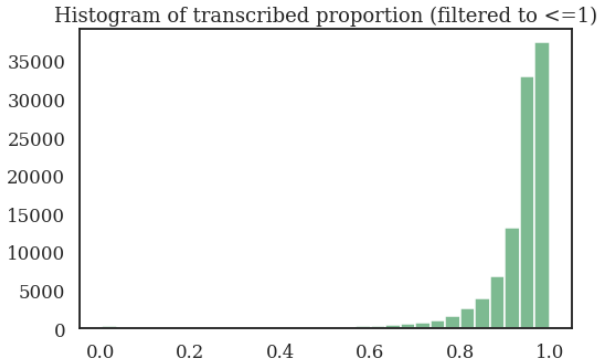

Figure 5: Distribution of transcribed proportion

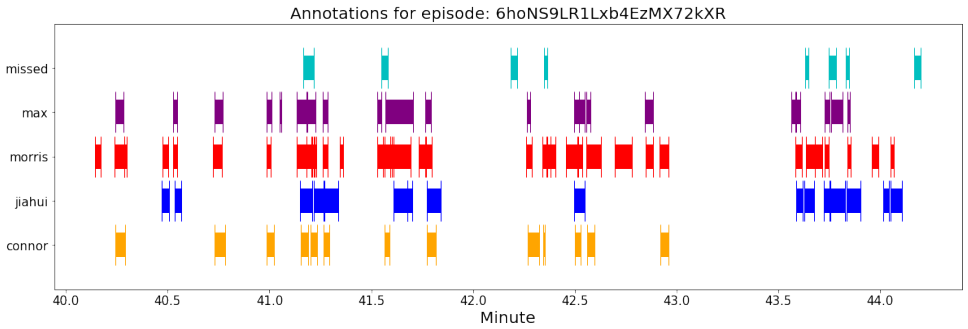

Figure 6: Comparison of utterances missed in transcripts vs. annotations

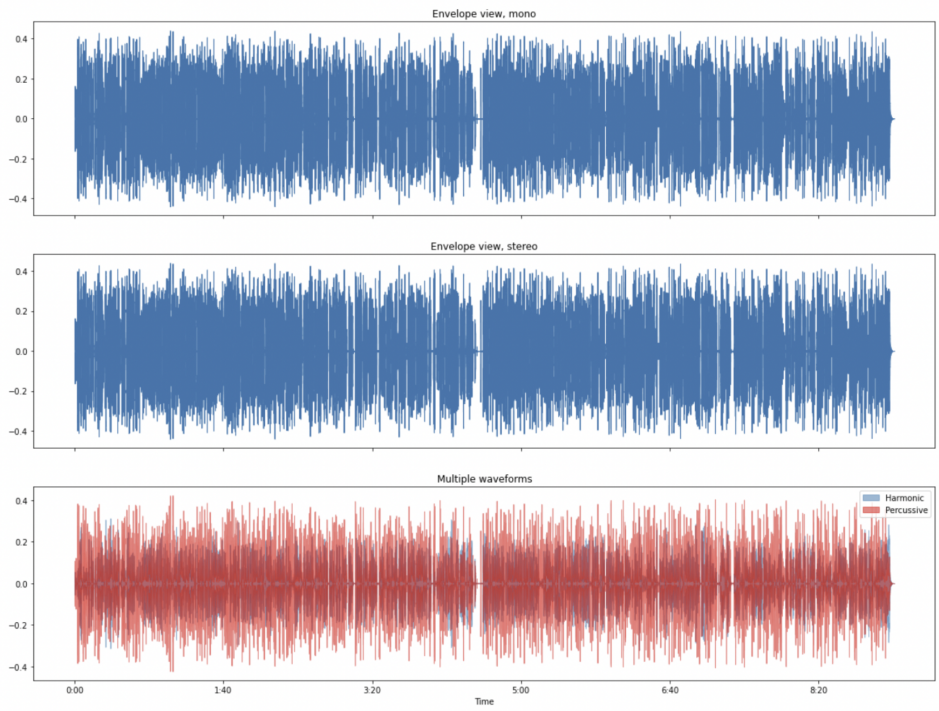

Figure 7: Envelope view of waveform

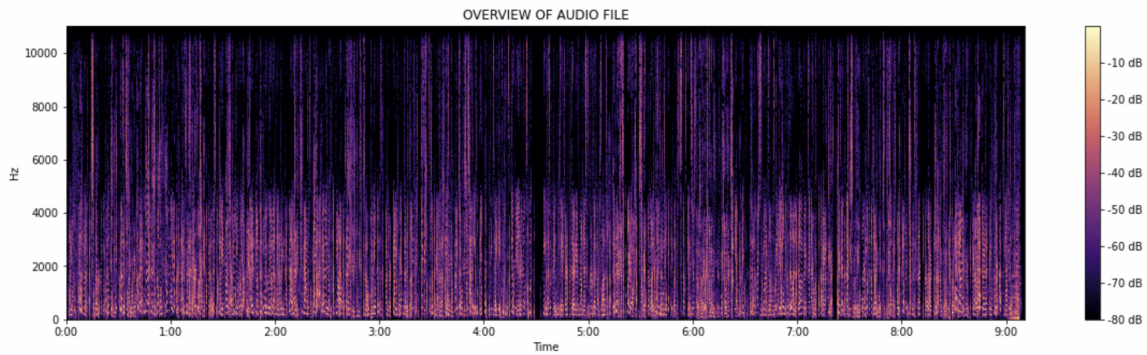

Figure 8: Time-frequency spectrogram

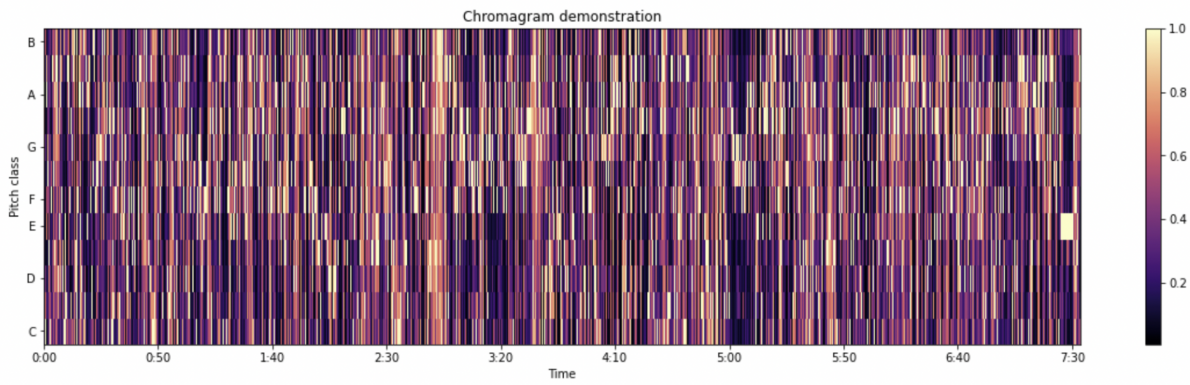

Figure 9: Chroma plot with pitch class intensity

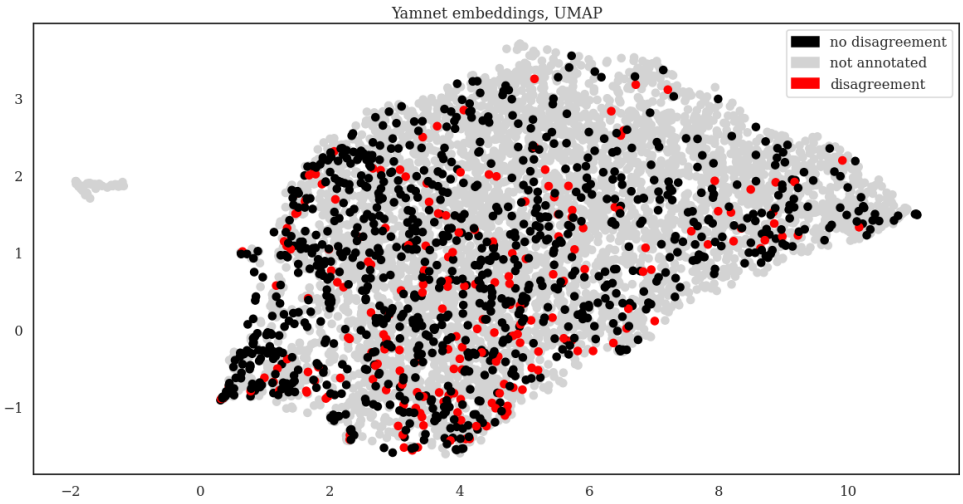

Figure 10: YAMNet Embeddings (UMAP Projection) for single annotated episode

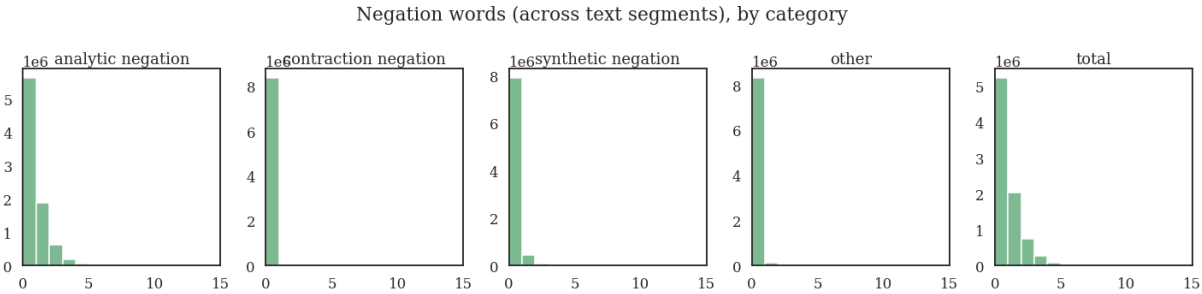

Figure 11: Distribution of negation words by category

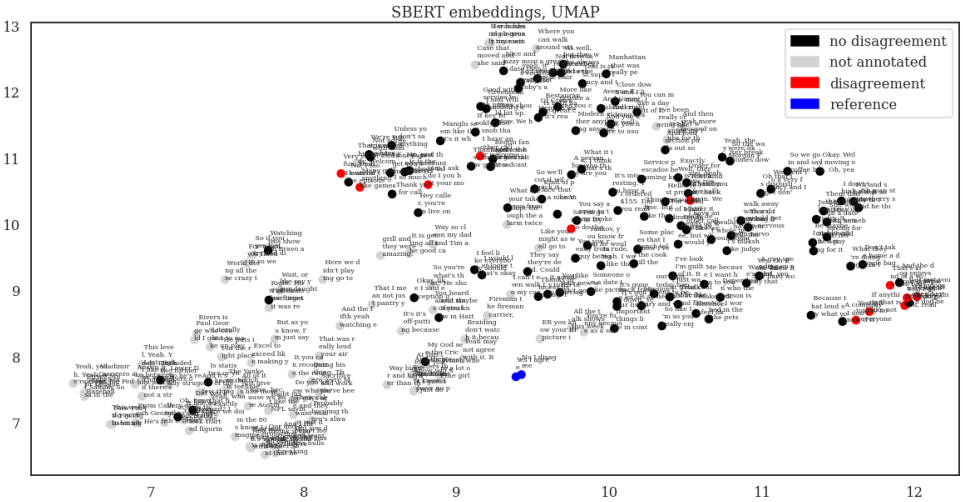

Figure 12: SBERT embeddings (UMAP projection), train and test episode

References

Ann Clifton, Aasish Pappu, Sravana Reddy, Yongze Yu, Jussi Karlgren, Ben Carterette, and Rosie Jones. 2020a. The spotify podcast dataset.

Ann Clifton, Sravana Reddy, Yongze Yu, Aasish Pappu, Rezvaneh Rezapour, Hamed Bonab, Maria Eskevich, Gareth Jones, Jussi Karlgren, Ben Carterette, and Rosie Jones. 2020b. 100,000 podcasts: A spoken English document corpus. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5903–5917, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Perry Cook George Tzanetakis, Georg Essl. 2001. Audio analysis using the discrete wavelet transform.

Ajda Gokcen and Marie-Catherine de Marneffe. 2015. I do not disagree: leveraging monolingual alignment to detect disagreement in dialogue. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 94–99, Beijing, China. Association for Computational Linguistics.

Dustin Hillard, Mari Ostendorf, and Elizabeth Shriberg. 2003. Detection of agreement vs. disagreement in meetings: Training with unlabeled data. In *Companion Volume of the Proceedings of HLT-NAACL 2003 - Short Papers*, pages 34–36.

Juan Pablo Bello Justin Salamon. 2016. Deep convolutional neural networks and data augmentation for environmental sound classification.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Lu Wang and Claire Cardie. 2014. Improving agreement and disagreement identification in online discussions with a socially-tuned sentiment lexicon. In *Proceedings of the 5th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 97–106, Baltimore, Maryland. Association for Computational Linguistics.

Wen Wang, Sibel Yaman, Kristin Precoda, Colleen Richey, and Geoffrey Raymond. 2011. Detection of agreement and disagreement in broadcast conversations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 374–378, Portland, Oregon, USA. Association for Computational Linguistics.

Chang Xu, Cecile Paris, Surya Nepal, and Ross Sparks. 2019. Recognising agreement and disagreement between stances with reason comparing networks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4665–4671, Florence, Italy. Association for Computational Linguistics.