Models Overview

Speaker change detection in general is a process of identifying timestamps within an audio recording where one speaker ends and another begins. Specifically under this context, OpenAI Whisper is applied first to automatically convert audio to transcriptions and generate segments within the audio file. However, Whisper segments are usually the smaller subsets of the speaker segment, which means that speaker identity may not change across sequential Whisper segments. Thus, the speaker change detection models below are applied to determine if a speaker changes from the previous Whisper segment to the next Whisper segment.

- PyAnnote

  It is the most popularly used speaker change detection package, which uses an audio-based speaker change detection model. It uses clustering methods based on merely audio features e.g. pitch, …, to detect speaker change.

- Spectral Clustering

  It is one of the most popular models to detect speaker change in research. It is also the audio-based speaker change detection model. It first converted audio files to vector embeddings, and then applied spectral clustering to the vector embeddings to detect speaker change.

- NLP Rule-based Model

  It is a new text-based speaker change detection model developed by us. It detects speaker changes through rule-based analysis with NLP.

  Upon closer examination of over hundreds of Whisper text segments, clear patterns exist in the Whisper segments so people could use them to identify that the speaker indeed changes across these text segments with almost 100% certainty. The Spacy NLP model is used to translate these identifiable patterns based on human comprehension into rules in programming languages. These rules are used to determine if the well-defined patterns exist in text segments to identify if a speaker changes across these segments. Specifically, the rules are below.

  - If the segment starts with the lowercase character, the segment continues the previous sentence. The speaker does not change in this segment.
  - If the sentence ends with ?, and its following sentence ends with . The speaker changes in the next segment.

    - If there is the conjunction word in the beginning of segment. The speaker does not change in this segment.

- Llama2 Model

  It is a new text-based speaker change detection model developed by us using large language models, specifically Llama2.
  For the text-based speaker change detection, no public models are available so this is an area of innovation. The Llama2-70b model is chosen to detect speaker change as it has the best reading comprehension performance among open-source models which could be run locally without additional charges. The prompts for speaker change detection were developed meticulously to ensure that Llama2 could understand the question, perform the speaker change detection of each segment, and return the answer in a standardized JSON format. Also, the parameters of the llam2 model were set to ensure results reproducibility.

- Ensemble Models

  The ensemble models are new audio-and-text based speaker change detection models developed by us. Existing popular speaker change detection models use merely audio features to detect speaker change. These models perform well for audio files which have rich audio features. However, some audio files have speakers with similar audio features such as pitch and frequency, which leads to the poor performance of these merely audio-based speaker change detection models.

  To address these issues, ensemble models detect speaker changes based on both audio and text features by ensembling the prediction results of the four models mentioned above. The ensembling method using both audio features and textual features to detect speaker change increases the model robustness across various input files.

  Two types of ensemble models are developed based on different methods of ensembling the prediction results of the three models, pyannote-audio, Llama2-70b, and spectral clustering models.

    - The Majority Model: The ensemble method is majority voting. The Majority model predicts the speaker change as true if the majority of models predict it as true.

    - The Unanimity Model: The ensemble method is unanimity voting. The Unanimity model predicts the speaker change as false only if all models predict it as false. This model reduces the probability of mispredicting true speaker changes as false. Thus, it reduces the probability of mixing segments of different speakers together, which is very helpful for subsequent speaker identification tasks.

The ensemble models correct the ensemble prediction result using the rule-based NLP analysis to get its final prediction. Specifically, the ensemble model predicts speaker change as true or false if the rule-based NLP analysis predicts that based on rules developed by human comprehension.

Metrics Overview

The speaker change detection models express their prediction results in terms of segments. Specifically, the model generates individual segments with their starting timestamps and ending timestamps. Speaker change remains the same within the individual segment but differs across the segments.

To evaluate the performance of the models above to detect speaker change, four metrics are used to compare segments and speaker change timestamps between ground truth and predictions.

Coverage and purity are metrics that compare prediction segments with true segments by considering the overlap between the two segments.

Coverage is computed for each true segment as the ratio of the duration of the intersection with the most co-occurring prediction segment and the duration of the true segment. For instance, the first true segment indicates that the speaker remains the same from 0 to 10. The second true segment indicates that the speaker remains the same from 10 to 16. The first prediction segment indicates that speaker change remains the same from 0 to 6. The second prediction segment indicates that the speaker remains the same from 6 to 10. The third prediction segment indicates that the speaker remains the same from 10 to 16. The first prediction segment covers 60% of the first true segment. The second prediction segment covers 40% of the first true segment. Since the first prediction segment covers a higher proportion of the true segment than the second prediction segment, the coverage of the first true segment is 60% instead of 40%. The coverage of the second true segment is 100% as the second true segment completely overlaps the third prediction segment. Thus, the higher coverage of the true segment is, the less proportion of the true segment is divided into various prediction segments.

Purity is computed for each prediction segment to indicate how pure the prediction segment is. In the same example above, the first prediction segment is 100% pure as the speaker remains the same during the whole prediction segment from 0 to 6. The second prediction segment indicates that the speaker remains the same from 6 to 16, while in reality the true segment indicates that the speaker changes at 10. Thus, the second prediction segment is 60% pure as the same speaker remains at most from 10 to 16, which is 60% of its total time length. The third prediction segment is 100% pure as the same speaker remains the same during the whole segment from 10 to 16. Thus, the higher coverage of the prediction segment is, the less proportion of the prediction segment is spoken by different speakers.

Coverage is computed for each true segment. Then, the final coverage of the input file is the duration-weighted average of the coverage of all true segments. Similarly, Purity is computed for each prediction segment. Then, the final purity of the input file is the duration-weighted average of the purity of all prediction segments. In the same example as above, the final coverage is 75%, which is the duration-weighted average of two true segments ((5*60%+3*100%)/8). The final purity is 90%, which is the duration weighted average of three predicted segments ((3*100%+2*60%+3*100%)/8).

Precision is defined as TruePositive/(TruePositive+FalsePositive). Under the speaker change context, the Positive is the predicted speaker change timestamps. Thus, Precision measures among all predicted speaker change timestamps, the proportion of predicted speaker change timestamps which are equal to the true speaker change timestamps within the threshold. The higher precision is, the less likely the model is to misdetect the false speaker change as true. In the same example as above, the true speaker change happens at 10 and 16. The predicted speaker change happens at 6, 10, and 16. The precision is 33% as among three predicted speaker change timestamps (6, 10, and 16), only two predicted speaker change timestamps (10 and 16) are true.

Recall is calculated as TruePositive/(TruePositive+FalseNegative). Precision and Recall have the same numerator, which are all predicted speaker change timestamps. In contrast, Recall measures among all true speaker change timestamps, the proportion of predicted speaker change timestamps which are equal to the true speaker change timestamps within the threshold. Thus, the higher recall is, the more likely the model is to detect true speaker changes when they indeed occur. In the same example as above, the recall is 100% as among two true speaker change timestamps (10 and 16), all of them are correctly predicted.

Please note that precision and recall would become zeroes when the dataset does not have any true positives (specifically no speaker changes under our case), which makes the metrics invalid under such a case.

The choice of metrics to evaluate models depends on the context. Under the context that it is important not to mix sounds of different speakers together to represent one speaker, purity and recall are extremely important. Under the context that it is important not to split the same speaker segment into different speaker segments, coverage and precision are very important. In the same example above, the purity and recall is 90% and 100% respectively, which indicates that prediction segments from the model are almost from the same speakers. This is consistent with the fact that two out of three prediction segments are completely spoken by the same speakers. The coverage and precision is 75% and 33% respectively, which is consistent with the fact that the true speaker segment from 0 to 10 is split into two prediction segments, 0 to 6 and 6 to 10.

Also, it should be noted that precision and recall are more appropriate metrics to measure the models performance under the imbalanced dataset, where the proportion of speaker changes is extremely low. Suppose under an extreme case where the speaker remains the same from 0 to

9 and changes once from 9 to 10 according to the ground truth, the segment coverage and purity are both 90% if the model predicts that the speaker remains the same from 0 to 10. However, the model fails to predict any true speaker change, which is not measured by coverage and purity. Precision and recall, as complements to coverage and purity, have zero values in this case, indicating the low prediction power of true speaker change of the model under the imbalance dataset where the proportion of speaker change is only 10%.

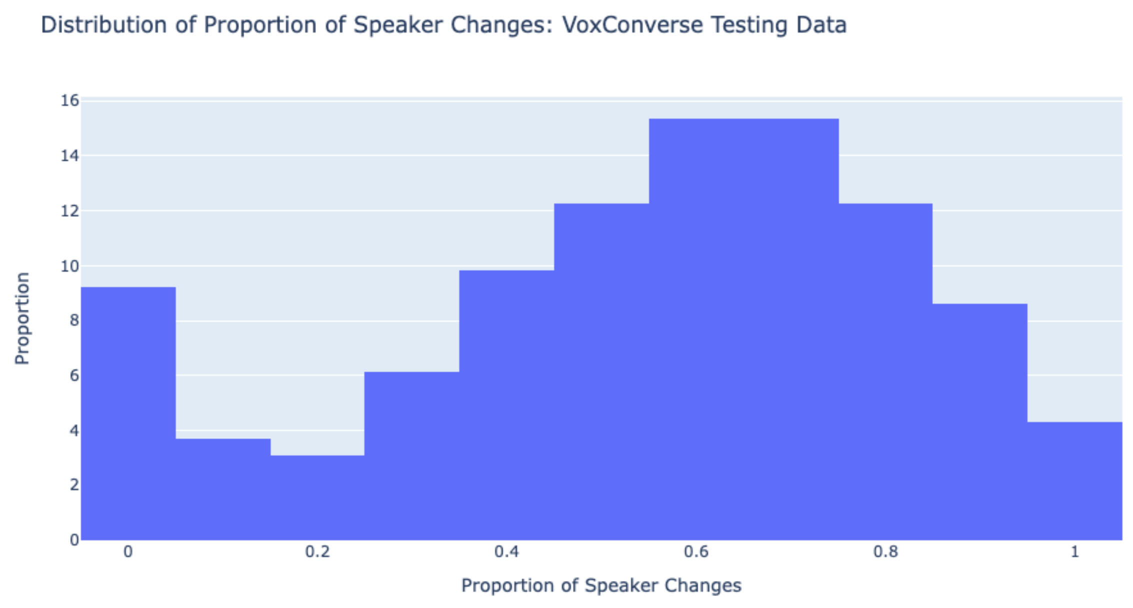
Evaluation Analysis

VoxConverse Dataset

Dataset Description

VoxConverse(v0.3)
VoxConverse is an only audio-visual diarisation dataset consisting of over 50 hours of multispeaker clips of human speech, extracted from YouTube videos, usually in a political debate or news segment context to ensure multi-speaker dialogue.

The audio files in the dataset have lots of variations of the proportion of speaker changes, which indicates the effectiveness of the dataset as the evaluation dataset to evaluate the models robustness.



Summary and Distribution Analysis

The VoxConverse has over 163 various files different in time length, the number of speakers, and the number of speaker changes. The metric of each model is first calculated for each file. Then, the average of metric values across files of the whole dataset is calculated for each model to compare the average performance across models. For each metric and each model, the proportion of files which have above or equal to the metric value threshold is calculated. The threshold of each metric value is selected as the lowest average metric value across all the models for both direct models comparison and distribution analysis. In addition, the distribution of each metric value for each model is plotted to ensure the analysis below is not affected by extreme outliers (see Appendix).

In terms of coverage, PyAnnote and the Majority model have higher performance than Llama2 the Unanimity based on both average statistics and the distributions. PyAnnote and the Majority model achieves around 85% average coverage and has more than 70% coverage values in above 85% input files.

In terms of purity, all models except for the Majority Model have above 80% average purity. In particular, Llama2 and the Unanimity model have more than 70% purity values in above 80% of the input files.

In terms of precision, all models have relatively low precision performance based on both the average statistics and the distributions. The Majority Model performs relatively better than other models in terms of precision.

In terms of recall, the Unanimity model has much better performance in recall than all other models based on average statistics and distribution analysis. Also, PyAnnote and the Majority model have extremely low recall performance as they achieve less than 19% precision values in more than 85% of files.

The choice of the model really depends on the particular use case and dataset. If the dataset characteristics are not clear or the use case puts nearly equal emphasis on the general model performance to detect both true speaker changes and false speaker changes with correct timestamps, the ensemble Audio-text-based Unanimity model should be used given it has best performance in recall, and consistently reasonable performance across all other metrics, coverage, purity, precision, and recall.

Moreover, the Ensemble Unanimity model should be used in the speech processing pipeline where the speaker change detection results would be used for subsequent tasks, including speaker diarization and speaker change detection given it has the highest purity and recall scores among all the models. The very low recall scores of PyAnnote indicates that it is very likely to misdetect or not detect true speaker change boundaries and thus mix segments of different speakers together into the segment of one speaker. Mistreating the segment spoken by different speakers as the segment spoken by one speaker would lower the performance of subsequent speaker related tasks after speaker change detection, such as speaker diarization and speaker identification. This suggests that the ensemble Unanimity model instead of PyAnnote should also be used under this case given its much higher precision scores.

Overall, without merely relying on the audio features, the ensemble Audio-text-based models use both Llama2 and NLP to identify speaker change by understanding the grammar structures and textual meaning across text segments. This comprehension of the textual features in addition to audio features significantly ensures the model robustness among all input files and consistently reasonable model performance across all the metrics.

Average Coverage, Purity, Precision, and Recall: VoxConverse Dataset

	PyAnnote	Llama2	Unanimity	Majority
Coverage	**86%**	45%	59%	**84%**
Purity	**83%**	**89%**	**87%**	70%
Precision	**23%**	14%	**24%**	**32%**
Recall	19%	**32%**	**41%**	19%

The Proportion of Input Files with Metric Value above or equal to the Lowest Performance Metric Value across All the Models: VoxConverse Dataset

	PyAnnote	Llama2	Unanimity	Majority
Coverage (45%)	85%	31%	53%	82%
Purity (70%)	72%	85%	83%	41%
Precision (14%)	22%	10%	22%	30%
Recall (19%)	12%	28%	37%	13%

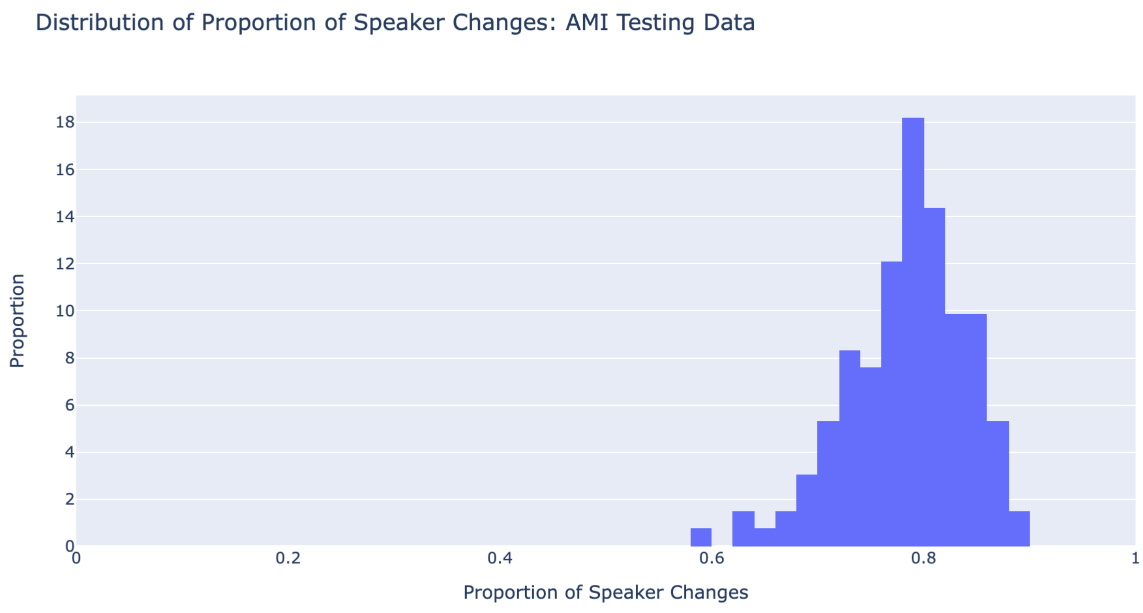
AMI Dataset

Dataset Description

AMI Headset Mix

The AMI Meeting Corpus is a multi-modal data set consisting of 100 hours of meeting recordings. Around two-thirds of the data has been elicited using a scenario in which the participants play different roles in a design team, taking a design project from kick-off to completion over the course of a day. The rest consists of naturally occurring meetings in a range of domains.

Different from VoxConverse Dataset, AMI dataset is not that diverse as it only consists of meeting recordings. The median and average proportion of speaker change is both around 78%, and the minimal proportion is above 59%. Thus, the evaluation analysis based on AMI is more applicable to measure the models performance under regular conversational settings.



Summary and Distribution Analysis

The summary and distribution analysis is conducted in the same way as analysis for VoxConverse Dataset before for direct comparison of model performance across datasets. The distribution plots in the Appendix are also checked to further confirmed that the analysis results are not affected by outliers.

Overall, the relative performance among models is same between VoxConverse and AMI Dataset while these two datasets differ both in the content, the proportion of speaker changes, and the diversification of audio files.

In terms of coverage, PyAnnote and the Majority model performs better than the Unanimity and Llama2 model in terms of both summary and distribution analysis.

In terms of purity, all models achieve relatively lower metrics of purity values than the VoxConverse dataset as the proportion of speaker changes increases. The Unanimity model and Llama2 model have higher purity value than the PyAnnote and the Majority model based on both summary and distribution analysis.

In terms of precision, all models except for Llama2 achieve over 40% purity on average. All models except for Llama2 has more than 32% purity in more than 35% of input files.

In terms of recall, the Unanimity model has the highest recall on average. In addition, it achieves recall scores of above 19% in the highest proportion of input files.

Thus, the Unanimity model is the most appropriate model under the business conversational setting in which the speaker change is above 60% given its consistent reasonable performance across coverage, purity, precision, and recall. This conclusion is consistent with the conclusion achieved by VoxConverse dataset, which further confirms the robustness of the Unanimity model across different input audio files characteristics.

Average Coverage, Purity, Precision, and Recall: AMI Dataset

	PyAnnote	Llama2	Unanimity	Majority
Coverage	**89%**	75%	80%	**92%**
Purity	**60%**	**65%**	**64%**	46%
Precision	**44%**	32%	**40%**	**46%**
Recall	18%	18%	**25%**	11%

The Proportion of Input Files with Metric Value above or equal to the Lowest Performance Metric Value across All the Models: AMI Dataset

	PyAnnote	Llama2	Unanimity	Majority
Coverage (75%)	89%	42%	60%	92%
Purity (46%)	58%	64%	60%	35%
Precision (32%)	41%	21%	36%	44%
Recall (11%)	17%	17%	24%	9%

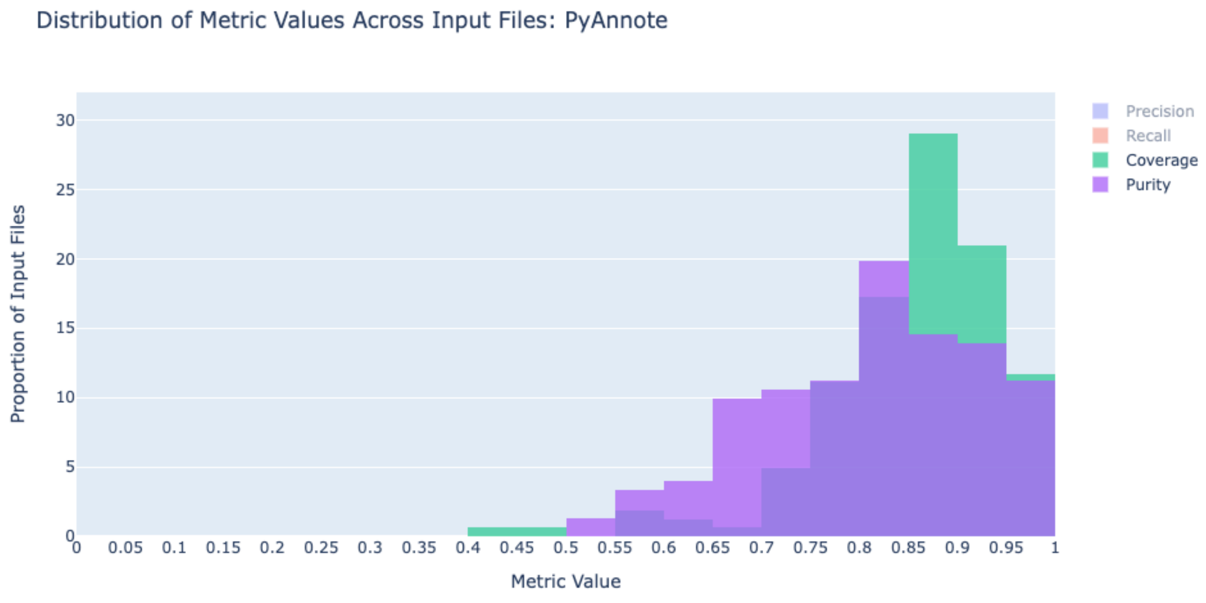
Appendix

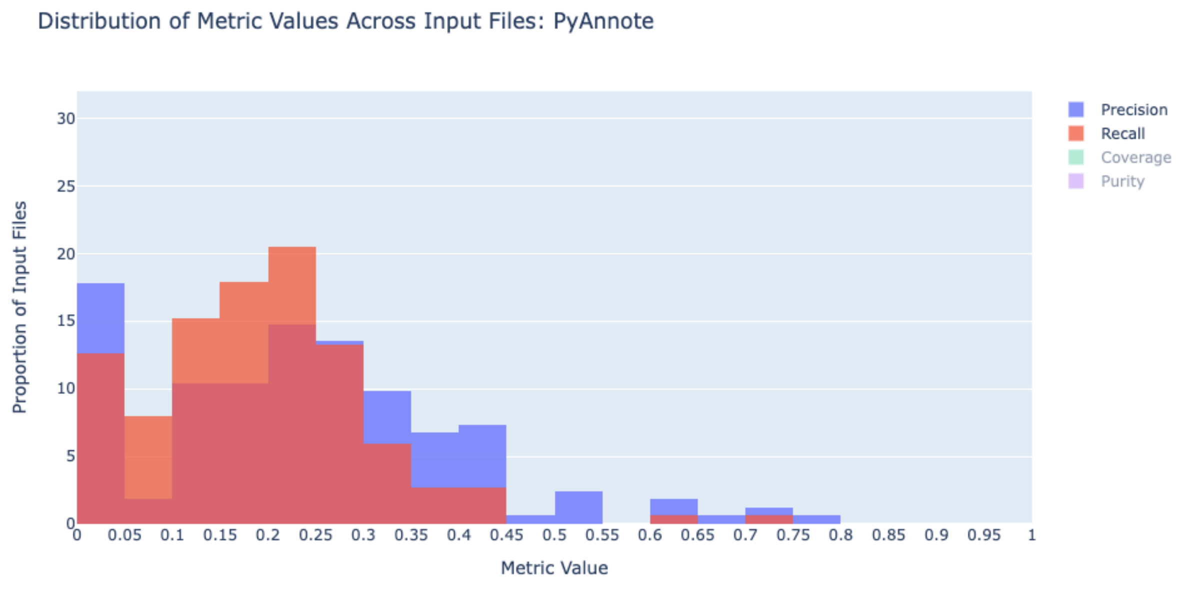
Distribution VoxConverse

The VoxConverse has over 163 various files different in time length, the number of speakers, and the number of speaker changes. The metric of each model is first calculated for each file.

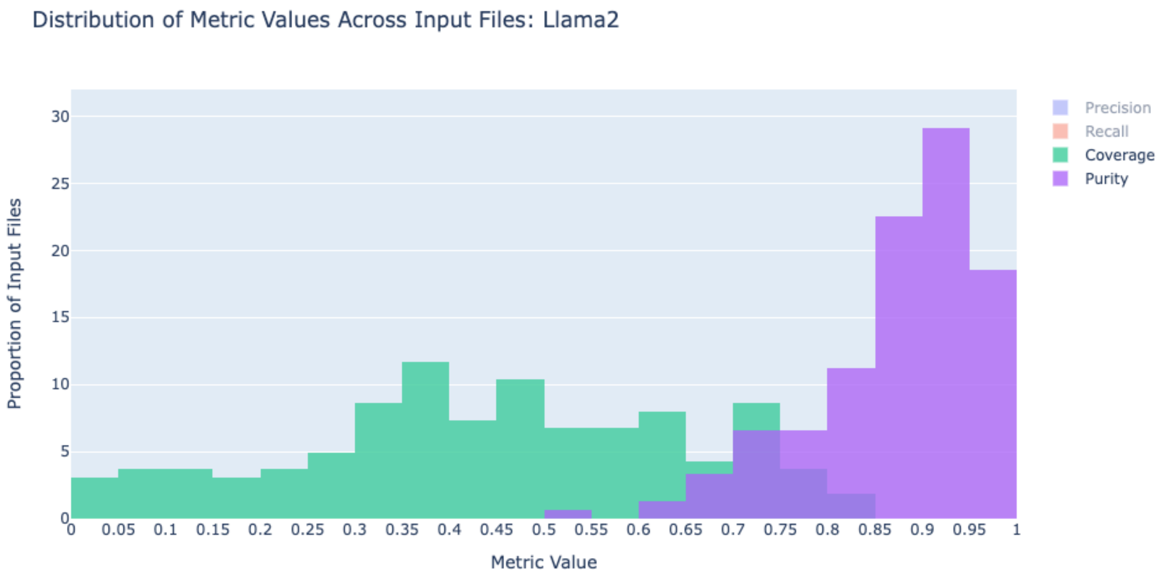
Then, the distribution of metric values across files of the whole dataset is plotted for each model below.

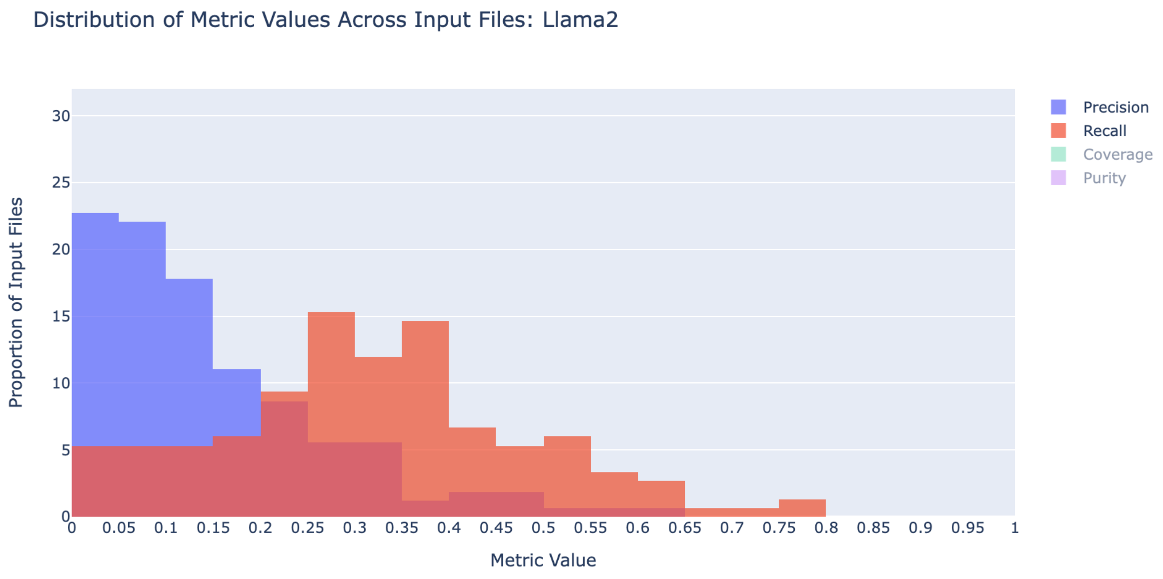
PyAnnote



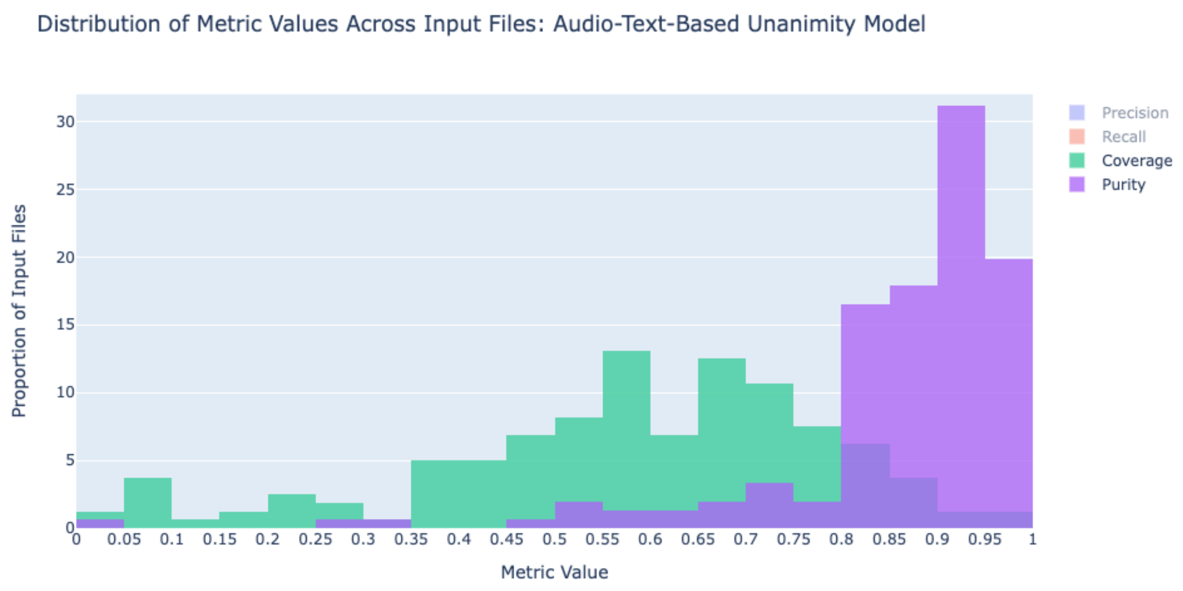


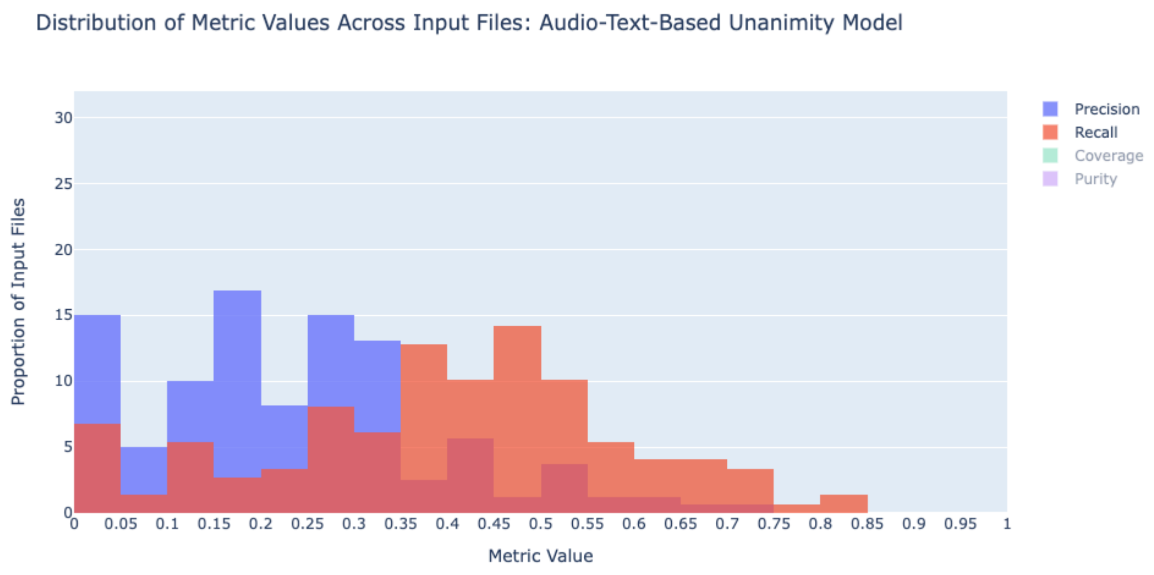
Llama2



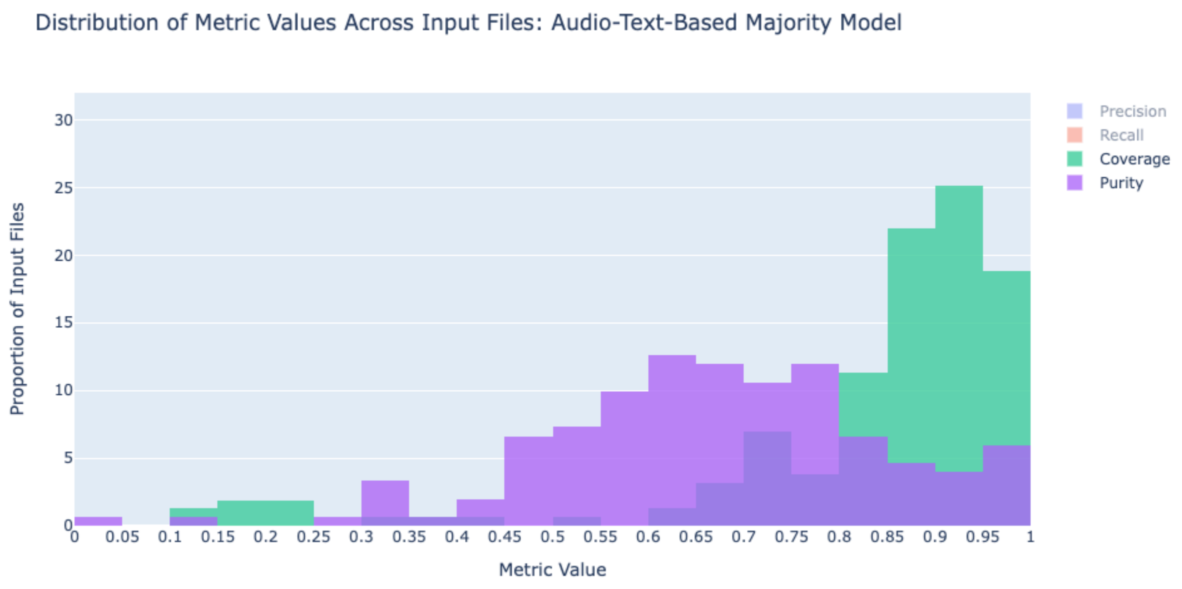


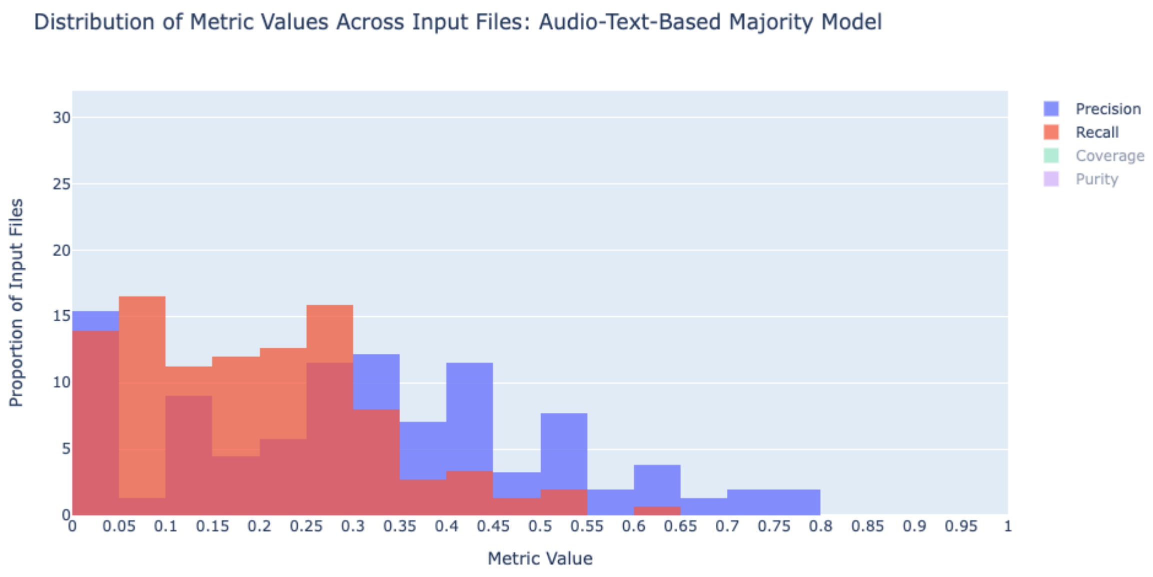
Audio-Text-Based Unanimity Model





Audio-Text-Based Majority Model

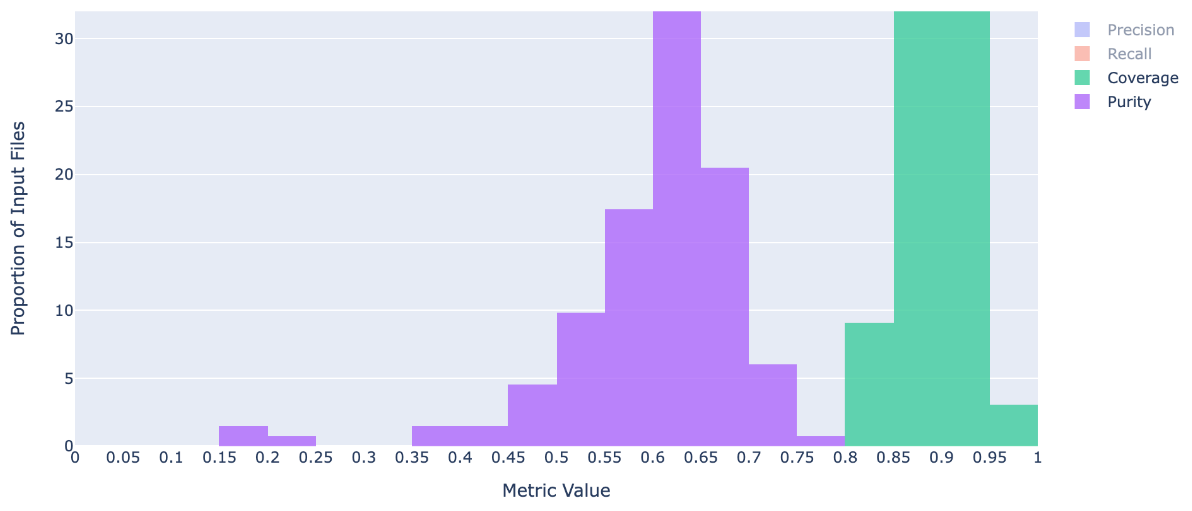




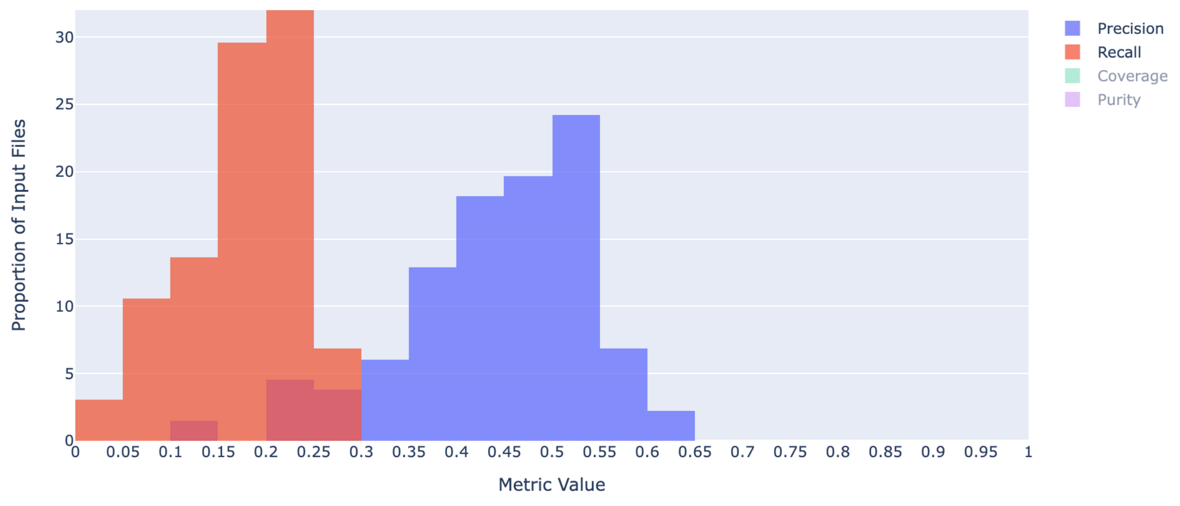
Distribution AMI

PyAnnote

Distribution of Metric Values Across Input Files: PyAnnote

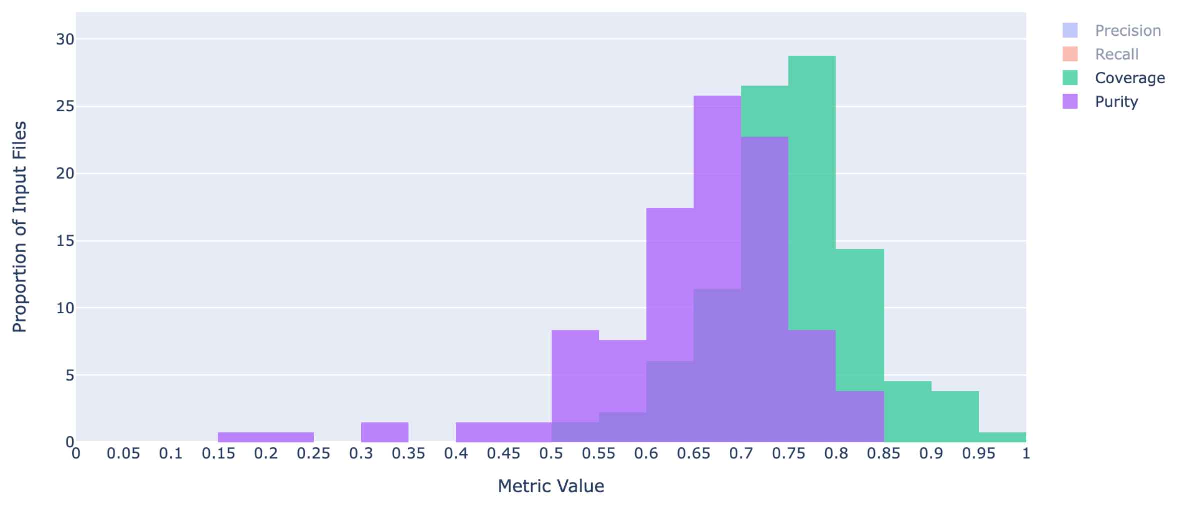


Distribution of Metric Values Across Input Files: PyAnnote

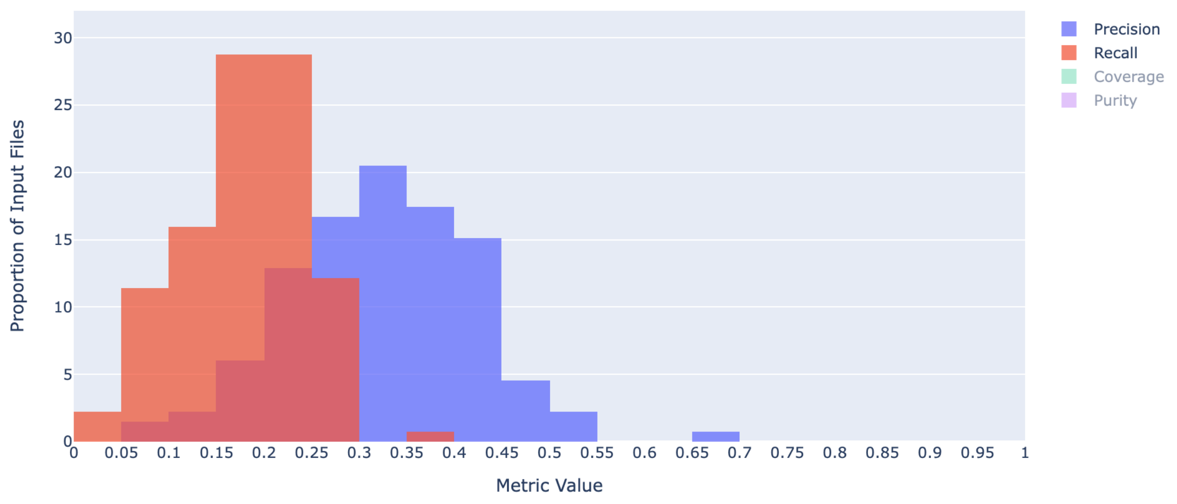


Llama2

Distribution of Metric Values Across Input Files: Llama2

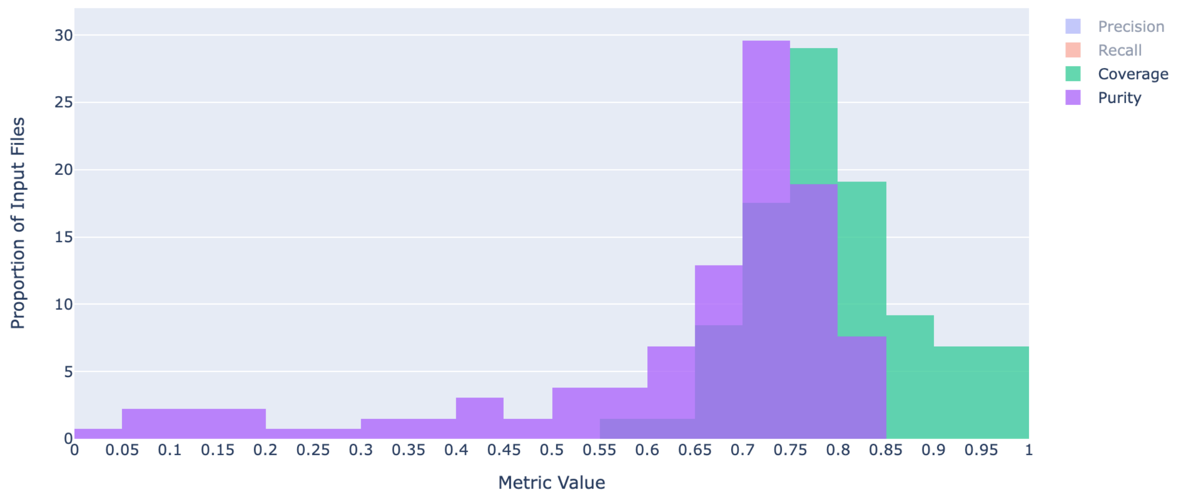


Distribution of Metric Values Across Input Files: Llama2

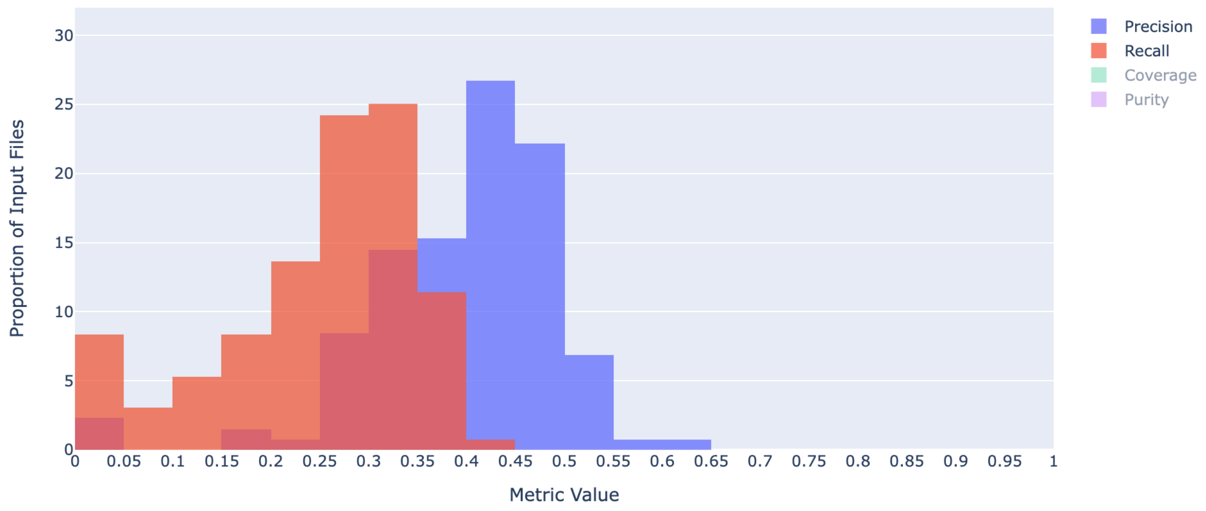


Audio-Text-Based Unanimity Model

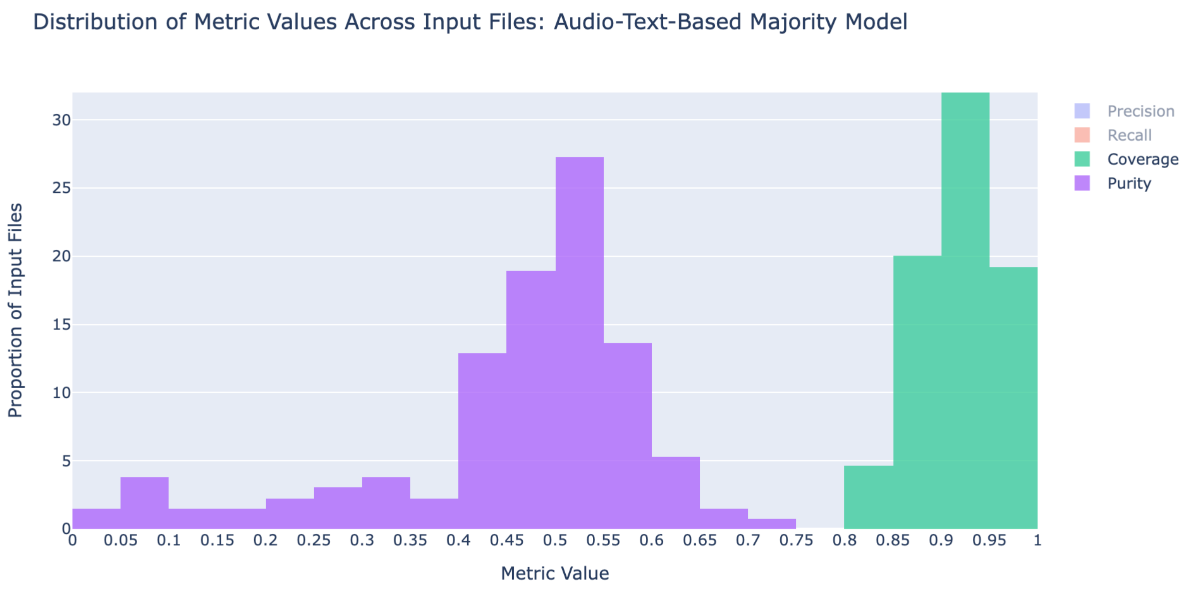
Distribution of Metric Values Across Input Files: Audio-Text-Based Unanimity Model

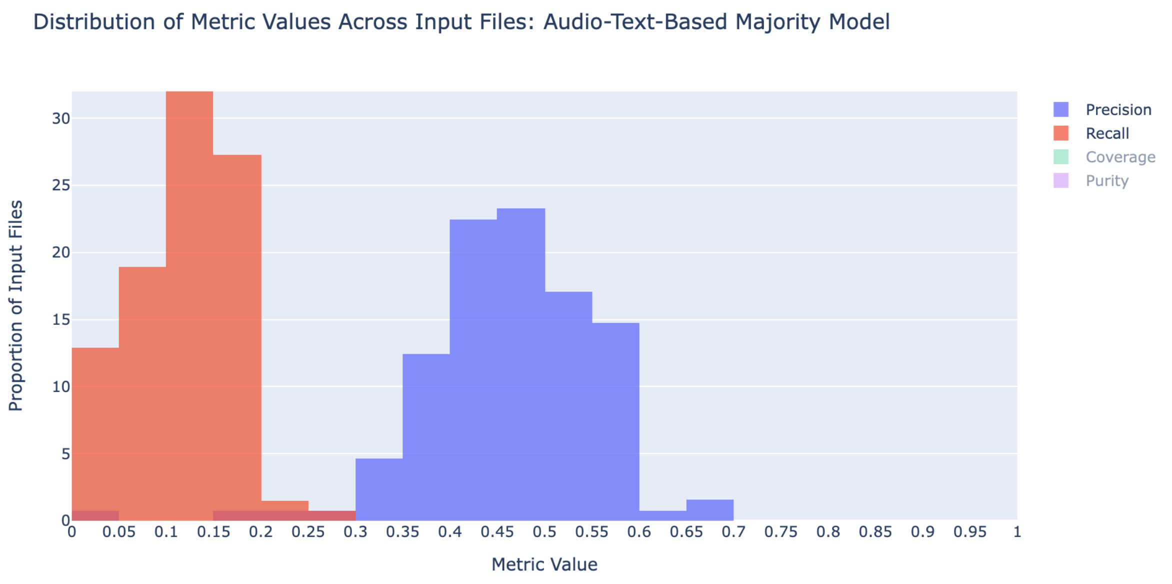


Distribution of Metric Values Across Input Files: Audio-Text-Based Unanimity Model



Audio-Text-Based Majority Model


Distribution of Metric Values Across Input Files: Audio-Text-Based Majority Model
Proportion of Input Files
Metric Value
Precision
Recall
Coverage
Purity
0
5
10
15
20
25
30
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0.5
0.55
0.6
0.65
0.7
0.75
0.8
0.85
0.9
0.95
1


Distribution of Metric Values Across Input Files: Audio-Text-Based Majority Model
Proportion of Input Files
Metric Value
Precision
Recall
Coverage
Purity
0
5
10
15
20
25
30
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0.5
0.55
0.6
0.65
0.7
0.75
0.8
0.85
0.9
0.95
1