INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

Query Based Acoustic Summarization for Podcasts

Samantha Kotey[1], *Rozenn Dahyot*[2], *Naomi Harte*[1]

[1]ADAPT Centre, School of Engineering, Trinity College Dublin, Ireland
[2]ADAPT Centre, Department of Computer Science, Maynooth University, Ireland

koteys@tcd.ie, rozenn.dahyot@mu.ie, nharte@tcd.ie

Abstract

Podcasts are a rich storytelling medium of long diverse conversations. Typically, listeners preview an episode through an audio clip, before deciding to consume the content. An automatic system that produces promotional clips, by supporting acoustic queries would greatly benefit podcasters. Previous text based methods do not use the acoustic signal directly or incorporate acoustic defined queries. Therefore, we propose a query based summarization approach, to produce audio clip summaries from podcast data. Leveraging unsupervised clustering methods, we apply our framework to the Spotify podcasts dataset. Audio signals are transformed into acoustic word embeddings, along with a pre-selected candidate query. We initiate the cluster centroids with the query vector and obtain the final snippets by computing a global and local similarity score. Additionally, we apply our framework to the AMI meeting dataset and demonstrate how audio can successfully be utilized to perform summarization.

Index Terms: query-based summarization, unsupervised speech summarization, clustering, acoustic word embeddings

1. Introduction

Podcast audio summarization is the task of generating a short audio clip which contains snippets of conversations in an episode. Due to the length and volume of episodes created, manually producing these promotional clips is a time consuming and tedious task. Podcasters require tools to help automate the process of creating audio teasers, that will entice listeners [1]. Researchers have approached the problem by developing models utilizing the Spotify podcasts dataset [2].

Typically, these systems rely on automatic speech recognition (ASR) to extract text representations, as input for multimodal [3], or text only models [4]. The audio summary is created by stitching together selected snippets [4], and evaluation is performed by comparing the audio-text pair to the original description, written by the podcast creator [5]. However, text based approaches to audio summarization are problematic in low resource real world scenarios, where ASR may not be available in different languages and transcripts contain errors [6].

While the use of ASR transcripts is hugely beneficial, the paralinguistic features of *'how'* something is said, can get lost in translation [7]. According to Martikainen et al. [8], users respond differently to podcasts based on voice characteristics such as tone and speech rate. In order to appeal to individual preferences, relevant audio clips should be identified to encourage listeners to consume more content [1]. An ideal system would support the podcaster to perform acoustic queries for contextually similar audio snippets [8], used to form a preview type summary.

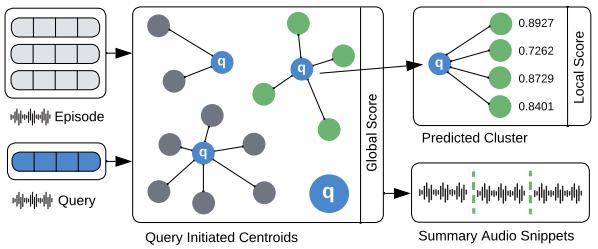

Figure 1: *Our proposed query based clustering framework. Clusters are initiated with the query and each embedding is assigned a local similarity score. A global score is obtained from the mean of the cluster's local scores. The predicted cluster has the maximum global score and snippets are used to create a summary.*

This two stage approach to extract salient sentences before summarizing the output has been implemented in query based text summarization [9]. The extraction stage involves supervised sentence selection, and summarization is performed by transformer models [10, 11]. External keywords or phrases can also be fed into end-to-end models as guidance signals [12, 13]. In acoustic models, guidance is in the form of spoken keywords, allowing users to query acoustic snippets [14].

Previous studies have shown improved performance using acoustic word embeddings (AWEs) [14], instead of dynamic time warping methods [15] for spoken query search. Transformer based acoustic models such as Wav2Vec2.0 [16] map raw audio signals to a fixed dimensional vector. Analogous to word embeddings, acoustic word embeddings represent similar sounding words, close together in an n-dimensional space. The distance between these vector representations can be measured, enabling the efficient implementation of vector search [14, 17].

The successful performance of AWEs can potentially be replicated in acoustic modeling tasks such as summarization. However, previous work has been limited, with few attempting methods without the direct use of text transcripts [18, 19, 20]. Recently, Wang [21] proposed a speech-to-speech summarisation model by applying traditional TF-IDF (term frequency-inverse document frequency) and clustering algorithms [22] to acoustic word embeddings in meeting conversations.

Inspired by Wang [21] and Settle [14], we propose an approach to extract audio clip snippets, constrained by a given audio query. As podcasts are a rich storytelling medium [1], we select named entities mentioned in a podcast conversation as the subject matter for the query. We hypothesize that queries representative of a named entity, will produce an audio summary focused on verbal utterances about that entity, with the same paralinguistic features. The contributions of this paper are as follows: firstly, we propose a query based approach to summa-

10.21437/Interspeech.2023-864

rize audio clips from the Spotify podcasts dataset. Leveraging unsupervised summarization methods, we devise a framework to cluster and score acoustic word embeddings, without guidance from ASR transcripts. Secondly, we conduct extensive experiments and apply our system to a different domain, on the AMI meeting summarization dataset [23]. The results show comparable results with previous work and demonstrates how audio can successfully be utilized to perform summarization.[1]

2. Proposed Approach

In this section, we introduce our unsupervised query based framework, as illustrated in Figure 1. First, we describe the processing steps to extract the candidate query and generate the acoustic word embeddings. Next, we detail the initialization process of the cluster centroids, and describe methods to determine the optimal number of clusters. Finally, we define different types of audio summaries, produced through global and local similarity scores.

2.1. Data Generation

2.1.1. Spotify Podcasts Dataset

The Spotify podcasts dataset [2] contains both professional and amateur created episodes, in the form of audio and transcription text. Each episode is summarized by a creator written description, employed as ground truth in summarization tasks [1]. We curate a subset of the 2020 testset, using the following selection criteria. We process the transcripts to create sentence level time stamps and remove any sentences with less than 4, or more than 20 words. This removes short sentences with stop words, and overly long sentences. The mean number of words per sentence was 15 with over 75% having more than 18 words. We select a maximum episode length of 300 sentences, so each episode is approximately the same duration. A name entity recognition library[2] is used to identify episodes with at least 6 of the same entities. The first sentence containing the entity is then used as the candidate query. For example, if an episode contains the entity *'Ellen'*, the episode is selected if *'Ellen'* appears more than 6 times in the transcript. This filtering process results in the selection of 189 out of 1027 episodes. The objective is to find episodes rich in name entities, to assist evaluation purposes.

2.1.2. AMI Dataset

The AMI meeting corpus is a widely used dataset, which comes with a supplied annotated testset of 20 meetings. In line with [21], the long abstractive testset summaries are used as ground truth. Different to podcasts, the AMI meetings are focused around topics such as *'designing a remote control'*. Therefore, we apply the RAKE[3] algorithm to extract high scoring keywords from each sentence in the corpus. The first sentence containing the keyword is used as the candidate query.

2.1.3. Acoustic Word Embeddings

We process the datasets to obtain the corresponding raw audio signal for each sentence and query in the podcast episode or meeting corpus. The raw audio signals are sampled at 16kHz and fed into a Wav2Vec2.0 model[4] to create high dimensional vectors with a fixed length of 768. The acoustic word embeddings that are generated at the last hidden layer of the Wav2Vec2.0 model, are the vector representations used in our clustering framework.

[1] https://github.com/sigmedia/qasumm

[2] https://huggingface.co/dslim/bert-base-NER

[3] https://pypi.org/project/rake-nltk/

[4] https://huggingface.co/facebook/wav2vec2-base-960h

2.2. K-means Clustering

We implement the k-means clustering algorithm to group together similar acoustic word embeddings into a given number of N clusters. Each cluster centroid $(C_1, \cdots, C_N)$ is defined by either a random vector [24] or a pre-defined query vector. We iterate over the embeddings $(j_1, \cdots, j_k)$ and assign each vector $\mathbf{j}$ to a cluster $\mathbf{C_j}$. The objective is to minimize the sum of squared distances between $\mathbf{q}$ cluster centroids and the acoustic vectors. Following Wang [21], we apply the mini-batch variant [25] of Lloyds k-means algorithm [26] to the processed datasets. Mini-batch k-means takes a random sample from a small batch of the acoustic word embeddings, to update the clusters at each iteration. Splitting the data into batches is proven to be computationally more efficient for large datasets [25], encouraging conversion at a faster rate. K-means also requires manually setting the value of N. We experiment with two methods [27, 28] to automatically determine the optimal N value for each episode or meeting. As the data is not uniform, the clusters formed can be non-globular in shape and have different densities. The silhouette score [27] measures the similarity between the vectors within a cluster, compared to the vectors in other clusters. DBCV [28] is suited to non-globular shaped clusters, and measures the density within clusters and between clusters.

2.3. Query Based Centroid Initialization

As shown by Gupta et al. [29], initialization is an important factor when dealing with high dimensional data. Randomly seeding each cluster with existing vectors from the data, improves the accuracy and speed of the clustering process [30]. Therefore, we implement three types of centroid initializations in our experiments; random, one and all.

- **Random:** We apply the Forgy method [24] to assign N centroids from randomly chosen vectors in the dataset.
- **One:** One cluster is initialized with a query vector $\mathbf{q}$ and the other centroid vectors $(N-1)$ are initialized randomly.
- **All:** Query vector $\mathbf{q}$, is replicated to initialize every cluster.

2.4. Query Based Cluster Prediction

After the clusters have been formed, we measure the cosine similarity between the query vector $\mathbf{q}$ and each acoustic embedding vector $\mathbf{j}$ in cluster $\mathbf{C_j}$. Each vector is assigned a local similarity score:

$$S_{\mathrm{L}_j}(\mathbf{q}, \mathbf{C_j}) = \frac{\mathbf{q} \cdot \mathbf{C_j}}{\|\mathbf{q}\| \|\mathbf{C_j}\|} \tag{1}$$

$$\mathbf{S}_{\mathrm{L}}(\mathbf{q}, \mathbf{C_j}) = (S_{\mathrm{L}_1}, \cdots, S_{\mathrm{L}_k}) \tag{2}$$

For each cluster, we compute the mean local similarity score to obtain a global similarity score for that cluster:

$$S_{\mathrm{G}_{C_j}} = \mathbb{E}[\mathbf{S}_{\mathrm{L}}(\mathbf{q}, \mathbf{C_j})] \tag{3}$$

$$\mathbf{S}_{\mathrm{G}} = (S_{\mathrm{G}_{C_1}}, \cdots, S_{\mathrm{G}_{C_N}}) \tag{4}$$

The cluster with the maximum global similarity score, is selected as the predicted cluster P_c:

$$P_c = \operatorname{argmax} \mathbf{S}_{\mathrm{G}} \tag{5}$$

2.5. Query Based Summarization

To create the summaries, the corresponding audio snippets to the acoustic embedding vectors are concatenated together (cf. Figure 1). However, in the absence of human evaluation, the text associated with the audio snippets start and end times, is compared against metrics. The top n words or sentences in chronological order, are taken for three types of summaries:

Optimal summary: Compared to ground truth descriptions, ROUGE-1 scores [31] are calculated for the top n corresponding words in each cluster. The text summary with the highest ROUGE-1 score is the best optimal summary for evaluation.

Predicted summary: The top n corresponding words from the predicted cluster are used for the evaluation summary. The predicted cluster has the highest global acoustic similarity score.

Top acoustic summary: In the predicted cluster, we take the top n corresponding sentences of the individual acoustic embeddings, with the highest local acoustic similarity score.

3. Experimental Setup

3.1. Implementation

To evaluate the summaries, we create three groups of experiments: pre-defined; silhouette; and DBCV. Within each group, we look at random, one and all initialization strategies. We experiment on two datasets, the Spotify podcasts dataset [2] and the AMI meeting corpus [23], and apply the k-means mini-batch implementation from the Scikit-learn library[5]. To begin, we pre-define *N=5*, and set range values of $N = \{5, 6, 7, 8, 9, 10\}$, for both silhouette [27] and DBCV [28]. The model iterates through the data points, creating clusters until the maximum value of 10 is reached. The cluster with the highest silhouette or DBCV score is selected as the optimal number of clusters to use. The maximum iteration parameter is set at 100, or early stopping when converged. The number of initialized centroids must always equal the number of clusters. If *N=5* and *init=random*, then all 5 cluster centroids are randomly taken from the data. If *N=5* and *init=one*, the query vector is assigned to one centroid, and the other four are initialized randomly. Finally, if *N=5* and *init=all*, each of the 5 clusters is seeded with the same query vector, replicated 5 times.

3.2. Evaluation Metrics

ROUGE-1 metrics [31] are used to evaluate the corresponding transcripts of the summary output against the ground truth. This metric is suitable as we compare uni-gram similarity of single words and not the longest common subsequence (ROUGE-L). We compare F1-scores, to determine a balanced average between precision and recall metrics. The podcast dataset was not specifically designed to evaluate audio summarization. Recently, TREC ran a competition to produce one minute audio summaries in 2021 [5]. The audio summary output was evaluated against abstractive creator descriptions. Another TREC competition in 2020 focused on text based abstractive summarization and these summaries were of a higher standard. Therefore, we compare our work to those participants [32, 33], with summary lengths ranging from 58 to 205 tokens. We limit the summary output for the optimal cluster and predicted cluster to 200 words, which is approximately 60 seconds in length. The top 10 corresponding sentences, in the top acoustic summary also produces the same audio duration. To make a fair comparison, we further divide the group of 189 episodes to extract 50 final episodes, with descriptions longer than 100 words. In the AMI dataset, we limit the summary output to 350 words, in line with [21]. The summaries are evaluated against the AMI long abstractive summaries with an average length of 296.6 tokens.

[5]https://scikit-learn.org

Query	Top acoustic summary
Lessons hadn't yet begun when **Ellen** was summoned from her room.	Lessons hadn't yet begun when **Ellen** was summoned from her room. She was bundled into a waiting carriage ... Ways into their journey to meet with **Ellen's** father Edward confessed something darker was afoot as **Ellen** listened patiently ... Happily engaged Edward and **Ellen** made for the Scottish Village of Gretna just over the Scottish English border ... There **Ellen** was finally told the truth about Edwards plot. Edward and his brother were both tried in Lancaster for the Abduction of **Ellen** Turner ...
We're trying to force **Takashi** into his car to kidnap him.	We're trying to force **Takashi** into his car to kidnap him. That's why he wanted to kidnap ... Shadi sent a text the day after **Takashi** performed in Philly. When it was all over **Takashi** went on Instagram live with his longtime girlfriend, Sarah Molina. Jim said that nitration make life tough for **Takashi** security team so that they would stop protecting him ...
Frank was close to both of his parents, but he had a special bond with his dad..	**Frank** was close to both of his parents, but he had a special bond with his dad. **Frank** was witnessing an entirely new side of his dad in the city... City the system was highly efficient as long as **Frank** avoided hitting the same cities or hotels twice. This mindset governed **Frank's** life whenever he saw an obstacle ...

Table 1: *Examples of top acoustic summaries from podcast data split 189/1027, showing the top 10 ordered corresponding sentences with the highest local acoustic similarity score.*

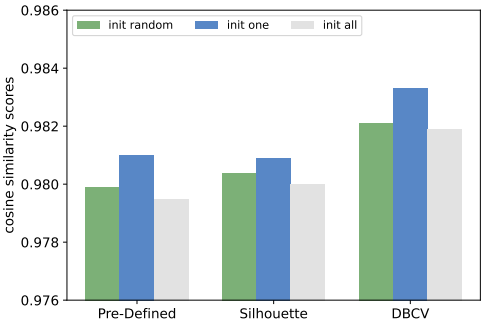

Figure 2: *Barchart depicting the mean global acoustic similarity scores of all predicted clusters, grouped by the cluster selection method, for podcast data split 189/1027.*

4. Results

In Table 1, we present the qualitative results of the top acoustic summary. The objective was to produce a summary constrained by the candidate audio query. Visually, we can see the named entity from the query is mentioned multiple times in the transcript summary. The clustering is shown to be effective as additional similarities between other sentences are present. While the summary is representative of the query, the narrative of the story is lacking context. This output is typical in extractive summaries, as conversations are disjointed in nature. Performing abstractive summarization on long passages of extracted text [5], helps to give meaning and context. Since our goal is to give listeners a short audio preview, our clustering framework achieves good results. Figure 2 presents the cosine similarity scores between the query vector and all of the audio vectors in the predicted cluster (global score). We compute a mean of the global similarity scores and group them to show differences between pre-defined, silhouette and DBCV methods. We note the cosine similarity scores are high, because each method successfully predicts the cluster containing the query vector. Next, in Table 2 and Table 3, we present the ROUGE-1 experiment results for the podcast and AMI datasets respectively.

PODCAST ROUGE-1(%)			
Method	**Precision**	**Recall**	**F1**
optimal summary			
pre-defined	26.96	32.27	28.14
dbcv	**27.25**	32.19	**28.18**
silhouette	27.03	**32.39**	28.17
predicted summary			
pre-defined + init (0)	23.51	27.98	24.18
pre-defined + init (1)	24.02	**29.11**	**25.00**
pre-defined + init (5)	25.01	27.71	24.25
silhouette + init (0)	24.25	27.80	24.40
silhouette + init (1)	24.20	28.53	24.69
silhouette + init (5)	25.08	27.58	24.18
dbcv + init (0)	25.06	25.16	22.74
dbcv + init (1)	24.00	26.74	23.26
dbcv + init (5)	**25.11**	25.65	22.96
top acoustic summary			
pre-defined + init (0)	26.19	20.51	22.67
pre-defined + init (1)	28.09	**21.34**	**23.22**
pre-defined + init (5)	**29.43**	20.76	22.75
cuedspeechUniv2 [32]	55.82	19.66	**27.94**
category-aware [33]	46.92	19.39	24.40

Table 2: *Precision, Recall & F1-Scores for PODCAST testset 50/1027. Denoted as; pre-defined (N=5), silhouette and dbcv (N=5-10) + init (0)=random, init (1)=one, init (5)=all.*

Centroid Initialization: We identify a trend in our initialization approach. Figure 2 shows that initializing with one query vector *(init=one)*, outperforms random *(init=random)*, and initializing with all query vectors *(init=all)* is less effective. A similar trend can be seen in both Table 2 and Table 3. Specifically, in the predicted summary (pre-defined + init 1), pre-defining the cluster *N=5*, and initializing the centroid with *init=one*, shows a higher F1-score of 25% compared to 24.18% for random initialization (pre-defined + init 0). As we are using high dimensional embedding features, giving the model a starting point will greatly improve the clustering formation. According to Arthur et al. [30], it is essential to ensure the initial centroids are far apart. This explains the reduced performance when the clusters are initialized with 5 of the same query vectors. A better approach would be to use 5 different queries to seed the clusters. However, this is outside the scope of this research study.

Optimal Cluster Selection: We observe that Figure 2 shows DBCV outperforming silhouette when measuring global acoustic similarity scores. However, in Table 2, DBCV achieves 23.26% F1-scores (dbcv + init 1), which is lower than silhouette at 24.69.% for method (silhouette + init 1). There is a clear difference when measuring acoustic similarity and text similarity. While running our experiments, we noted that DBCV favours a higher number of dense clusters (max=10) in the podcast data, resulting in a smaller number of sentences per cluster for evaluation. In Table 3, for the AMI dataset, the DBCV method performs better (dbcv + init 1), compared to pre-defining cluster values (pre-defined + init 1). To effectively compare DBCV and silhouette, we set the same range score for both datasets (5-10). However, as each episode or meeting is unique, individual ranges would have produced better results.

Summary Types: In Table 2, we compare our three summary types to the TREC 2020 participants. The **optimal summary** (dbcv) outperforms Manakul and Gales [32], verifying the audio summaries have substance and contain quality material. In Table 3, the optimal summary (dbcv) on the AMI dataset outperforms Wang [21] with an F1-score of 35.78%. These scores demonstrate our system can successfully be generalized for different domains to cluster acoustic signals. In Table 2, the F1-score for the best podcast **predicted summary** (pre-defined + init 1) is 25%, which is lower than *cuedspeechUniv2* at 27.94%, but higher than *category-aware* at 24.40%. Considering the summary is focused on a named entity query, the performance is more than sufficient. We note in Table 3, the best predicted summary for the AMI dataset (dbcv + init 1) scores 32.57%, lower than the baseline *ESSumm* at 34.96%. All methods in the **top acoustic summary** for both podcast and AMI achieve lower F1-scores compared to optimal and predicted summaries. A possible explanation is that descriptions which are used for comparison with ROUGE are not written to reflect our candidate query. The acoustic summaries are enriched with speech queries, which are difficult to reflect without human evaluation.

AMI ROUGE-1(%)			
Method	**Precision**	**Recall**	**F1**
optimal summary			
pre-defined	32.91	37.34	34.55
dbcv	**34.93**	**38.00**	**35.78**
silhouette	32.83	37.28	34.47
predicted summary			
pre-defined + init (0)	29.54	33.68	31.04
pre-defined + init (1)	**31.22**	33.56	31.63
pre-defined + init (5)	30.65	34.31	31.89
silhouette + init (0)	29.57	33.95	31.17
silhouette + init (1)	30.42	34.70	32.00
silhouette + init (5)	29.81	33.52	30.92
dbcv + init (0)	30.95	33.94	31.81
dbcv + init (1)	31.08	**35.33**	**32.57**
dbcv + init (5)	30.18	33.50	30.99
top acoustic summary			
pre-defined + init (0)	**34.97**	21.22	25.55
pre-defined + init (1)	34.84	**22.10**	**26.27**
pre-defined + init (5)	35.04	21.36	25.81
ESSumm [21]	31.63	**40.36**	34.96

Table 3: *Precision, Recall & F1-Scores for AMI testset 20/20. Denoted as; pre-defined (N=5), silhouette and dbcv (N=5-10) + init (0)=random, init (1)=one, init (5)=all.*

5. Conclusions

We have presented an unsupervised clustering framework to produce audio clip summaries, directly from raw audio queries. Leveraging acoustic word embeddings, we initiated the cluster centroids with an acoustic query, to encourage enhanced cluster formations. We experimented with methods to select the optimal number of clusters and produced three types of summaries, through our global and local scoring system. Our approach demonstrated the efficacy of producing extractive summaries, acoustically similar to an acoustic query. We demonstrate the versatility of our framework by applying the system to podcasting and meeting data. In future work we plan to experiment with both single and multiple word queries, to further enhance the performance of our system.

6. Acknowledgments

This research was conducted with the financial support of Irish Research Council (IRC) under Grant Agreement GOIPG/2019/2353 and the ADAPT SFI Research Centre under Grant No. 13/RC/2106_P2.

7. References

[1] R. Jones, H. Zamani, M. Schedl, C.-W. Chen, S. Reddy, A. Clifton, J. Karlgren, H. Hashemi, A. Pappu, Z. Nazari, L. Yang, O. Semerci, H. Bouchard, and B. Carterette, "Current challenges and future directions in podcast information access," in *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2021, pp. 1554–1565.

[2] A. Clifton, S. Reddy, Y. Yu, A. Pappu, R. Rezapour, H. Bonab, M. Eskevich, G. Jones, J. Karlgren, B. Carterette, and R. Jones, "100,000 podcasts: A spoken english document corpus," in *Proceedings of the 28th International Conference on Computational Linguistics*, 2020, pp. 5903–5917.

[3] L. Vaiani, M. La Quatra, L. Cagliero, and P. Garza, "Leveraging multimodal content for podcast summarization," in *Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing*, 2022, pp. 863–870.

[4] A. Vartakavi, A. Garg, and Z. Rafii, "Audio summarization for podcasts," in *2021 29th European Signal Processing Conference (EUSIPCO)*, 2021, pp. 431–435.

[5] J. Karlgren, R. Jones, B. Carterette, A. Clifton, M. Eskevich, G. J. Jones, S. Reddy, E. Tanaka, and M. I. Tanveer, "TREC 2021 podcasts track overview," in *Proceedings of the Thirtieth Text Retrieval Conference (TREC)*. NIST Special Publication, 2021.

[6] K. McKeown, J. Hirschberg, M. Galley, and S. Maskey, "From text to speech summarization," in *Proceedings. (ICASSP '05). IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2005.

[7] L. Yang, Y. Wang, D. Dunne, M. Sobolev, M. Naaman, and D. Estrin, "More than just words: Modeling Non-Textual characteristics of podcasts," in *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, 2019, pp. 276–284.

[8] K. Martikainen, J. Karlgren, and K. P. Truong, "Exploring audio-based stylistic variation in podcasts," in *Proc. Interspeech*, 2022.

[9] M. Zhong, D. Yin, T. Yu, A. Zaidi, M. Mutuma, R. Jha, A. Hassan, A. Celikyilmaz, Y. Liu, X. Qiu, and Others, "QMSum: A new benchmark for query-based multi-domain meeting summarization," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2021, pp. 5905–5921.

[10] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "BART: Denoising Sequence-to-Sequence pre-training for natural language generation, translation, and comprehension," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 7871–7880.

[11] I. Beltagy, M. E. Peters, and A. Cohan, "Longformer: The Long-Document transformer," *arXiv preprint arXiv:2004.05150*, 2020.

[12] J. He, W. Kryscinski, B. McCann, N. Rajani, and C. Xiong, "CTRLsum: Towards generic controllable text summarization," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 2022, pp. 5879–5915.

[13] Z.-Y. Dou, P. Liu, H. Hayashi, Z. Jiang, and G. Neubig, "GSum: A general framework for guided neural abstractive summarization," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2021, pp. 4830–4842.

[14] S. Settle, K. Levin, H. Kamper, and K. Livescu, "Query-by-Example search with discriminative neural acoustic word embeddings," *Proc. Interspeech*, 2017.

[15] T. J. Hazen, W. Shen, and C. White, "Query-by-example spoken term detection using phonetic posteriorgram templates," in *2009 IEEE Workshop on Automatic Speech Recognition & Understanding*, Dec. 2009, pp. 421–426.

[16] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "Wav2vec 2.0: A framework for self-supervised learning of speech representations," *Adv. Neural Inf. Process. Syst.*, vol. 33, pp. 12 449–12 460, 2020.

[17] Y. Yuan, L. Xie, C.-C. Leung, H. Chen, and B. Ma, "Fast Query-by-Example speech search using Attention-Based deep binary embeddings," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 1988–2000, 2020.

[18] X. Zhu, G. Penn, and F. Rudzicz, "Summarizing multiple spoken documents: finding evidence from untranscribed audio," in *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, 2009, pp. 549–557.

[19] S. Maskey and J. Hirschberg, "Summarizing speech without text using hidden markov models," in *Proceedings of the human language technology conference of the NAACL, companion volume: short papers*, 2006.

[20] R. Flamary, X. Anguera, and N. Oliver, "Spoken WordCloud: Clustering recurrent patterns in speech," in *2011 9th International Workshop on Content-Based Multimedia Indexing (CBMI)*, 2011, pp. 133–138.

[21] J. Wang, "ESSumm: Extractive Speech Summarization from Untranscribed Meeting," in *Proc. Interspeech*, 2022, pp. 3243–3247.

[22] M. Bilenko, S. Basu, and R. J. Mooney, "Integrating constraints and metric learning in semi-supervised clustering," in *Proceedings of the twenty-first international conference on Machine learning*, 2004, p. 11.

[23] J. Carletta, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, W. Kraaij, M. Kronenthal, G. Lathoud, M. Lincoln, A. Lisowska, I. McCowan, W. Post, D. Reidsma, and P. Wellner, "The AMI meeting corpus: A pre-announcement," in *Machine Learning for Multimodal Interaction*, 2006, pp. 28–39.

[24] E. W. Forgy, "Cluster analysis of multivariate data: Efficiency vs. interpretability of classifications," *Biometrics*, vol. 21, pp. 768–769, 1965.

[25] D. Sculley, "Web-scale k-means clustering," in *Proceedings of the 19th international conference on World wide web*, 2010, pp. 1177–1178.

[26] S. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Inf. Theory*, vol. 28, no. 2, pp. 129–137, 1982.

[27] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Journal of computational and applied mathematics*, pp. 53–65, 1987.

[28] D. Moulavi, P. A. Jaskowiak, R. J. G. B. Campello, A. Zimek, and J. Sander, "Density-based clustering validation," in *Proceedings of the 2014 SIAM International Conference on Data Mining*, 2014.

[29] A. Gupta, A. Tomer, and S. Dahiya, "Importance of initialization in K-Means clustering," in *2022 Second International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT)*, 2022, pp. 1–7.

[30] D. Arthur and S. Vassilvitskii, "K-Means++: The advantages of careful seeding," in *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, 2007, pp. 1027–1035.

[31] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," in *Proceedings of the ACL-04 Workshop on Text Summarization Branches Out.*, 2004, pp. 74–81.

[32] P. Manakul and M. Gales, "CUED_SPEECH at TREC 2020 podcast summarisation track," in *Proceedings from the 29th Text Retrieval Conference (TREC)*. NIST, 2020.

[33] R. Rezapour, S. Reddy, A. Clifton, and R. Jones, "Spotify at TREC 2020: Genre-Aware abstractive podcast summarization," in *Proceedings from the 29th Text Retrieval Conference (TREC)*. NIST, 2020.