Universität Leipzig
Text Mining and Retrieval
Degree Programme Computer Science and Media

Audio- and text-based Podcast Retrieval and Summarization

Master's Thesis

Jakob Schwerter

1. Referee: Junior-Prof. Dr. Martin Potthast
2. Referee: Dr. Andreas Niekler

Submission date: February 11, 2022

Declaration

Unless otherwise indicated in the text or references, this thesis is entirely the product of my own scholarly work.

Leipzig, February 11, 2022

.....
Jakob Schwerter

Abstract

This thesis proposes approaches for two fields in the domain of podcasts, re-ranking topical retrieval results of fixed two-minute podcast segments and automatic textual summarization of podcast episodes.

Regarding segment retrieval, we show the effect of the additional usage of audio data in contrast to only a transcription of the podcast. Therefore, we propose and compare multiple approaches for the re-ranking of podcast segments based on three criteria. The criteria are whether a segment is entertaining, whether it contains opinions and whether it contains discussion. One approach utilizes only text data, another approach uses only audio data and a third approach incorporates a combination of both.

Two approaches are proposed for summarization. One approach produces abstractive summaries and utilizes a DistilBART summarization model. The other approach produces extractive summaries and is based on the TextRank algorithm. Summaries by both systems are generated by prioritizing entertaining segments of the podcast. Both approaches generate short text summaries for podcast episodes intended to arouse interest of potential listeners.

Contents

Acknowledgements

Computations for this work were done (in part) using resources of the Leipzig University Computing Centre.

Chapter 1

Introduction

Podcasts have become an increasingly popular form of media in recent years. According to Edison Research [2020], 37% of Americans above the age of 12 (approximately 104 million people) listened to podcasts monthly in 2020. A 5% increase from 32% in 2019 and continuing the growth trend they have been measuring since 2009. And with 48 million published podcast episodes as of April 2021 [Winn, 2021], a number that increases daily, listeners have a plethora of offerings to choose from. However, such an expansive selection of content can easily become a burden to podcast listeners. A sample by Misener [2019] of almost 19 million episodes published between 2005 and 2019 showed a median length of 36 minutes and 34 seconds. Therefore, listening to even a single podcast episode constitutes a rather significant commitment of time from the listener and trying to decide whether an episode is worth this commitment can be a challenging task. Furthermore, only certain parts of podcast episodes may be relevant to the interests of some podcast listeners. This thesis aims to aid podcast listeners in their podcast episode selection.

Additionally, this thesis is motivated by the podcast track of the Text Retrieval Conference (TREC),[1] which organizes two shared tasks for podcast segment retrieval and podcast summarization. It was organized for the first time in 2020 by Jones et al. [2020]. This thesis proposes approaches for both tasks.

The first shared task consisting of the retrieval and re-ranking of relevant podcast snippets is motivated as follows. Only listening to certain snippets of an episode (e.g. an excerpt on a specific topic) might be of interest to podcast listeners to save time by ignoring parts they are not interested in. For example, a listener might only want to listen to podcast snippets about the United States presidential election in 2020 or a newly released movie. However, podcasts are

[1] `https://trecpodcasts.github.io/`

usually published without a transcription of the content. This means that the content of the episodes can not be easily indexed by common search engines. A simple search to retrieve a particular segment in an episode or finding episodes that address specific topics, two ubiquitous tasks with regard to text based content, is not yet widely available. Furthermore, users might want to search for episodes based on criteria that are not rooted in the subject matter. For example, a user might only be interested in episodes containing discussion between multiple people instead of one person talking alone. Or they might only want to listen to episodes that are intended as entertainment, such as comedy shows, in contrast to informational content. Therefore, this thesis proposes an approach to retrieve podcast snippets based on search queries and re-rank the results based on multiple criteria.

The second shared task consisting of the automatic summarization of podcast episodes is motivated as follows. For their first listen, a new listener only has limited information about an episode available, mostly the title and description, which are both set by the podcast creator. In some cases these might be helpful, but there exist only suggestions [Dennis, 2021] and no clearly defined rules for their format and content. Titles may only contain the date on which the episode was published (e.g. the episode "Monday Morning Podcast 10-11-21" by the "Monday Morning Podcast"[2]) or only episode numbers and names of guests (e.g. the episode "Episode 1265 B.J. Novak" by the "WTF with Marc Maron Podcast"[3]). Similar problems arise with the descriptions, which may only contain social media links or may only contain basically meaningless text (e.g. the episode "Charlie Gets Crippled" by "The Always Sunny Podcast"[4], which features the description "Yeah, yeah, yeah. Not that though."). Furthermore, the amount of times a podcast and its episodes are downloaded is a significant metric for creators, as it influences the ranking positions on popular platforms like Apple Podcasts and Spotify.[5] A higher position in the ranking leads to higher visibility and a higher chance of discovery by potential new listeners. Additionally, the amount of downloads is integral to the earnings a podcast can generate for its creator as included ads are usually paid for on the basis of a CPM (cost per mille) model, which measures advertising cost based on audience numbers [McLean, 2021]. The podcast creators are therefore incentivized to choose titles and descriptions that push potential listeners to download their content to earn more money. Subsequently, the titles and descriptions may not always be unbiased accounts of the episode's contents and may be sensationalized to draw the interest of

[2] https://open.spotify.com/episode/66aKvr6VlUImXXehWW6b4L
[3] https://open.spotify.com/episode/2DEZipWt12iQmJfd5XWZ76
[4] https://open.spotify.com/episode/1XKhUv0cMVdgI6ZXIH6ACy
[5] https://podcastcharts.byspotify.com/

more people. A concise and unbiased summary of the contents of an episode in the description is therefore not available for all episodes. But having such a summary at their disposal might benefit listeners who are trying to decide which episode they want to listen to or whether they want to listen to a particular episode. However, as shown by a survey published by Chan-Olmsted and Wang [2020], two of the most important motivators for podcast consumption are information and entertainment. We therefore aim to produce summaries that not only describe the content of episodes, but also still integrate entertaining aspects of the episodes. Therefore, this thesis proposes two approaches to automatically generate short text summaries for podcast episodes which prioritize entertaining segments.

There are several use cases for the proposed systems of both tasks. Search engines could integrate podcast retrieval to let users search for information in podcast episodes just like news content or pictures. Automatically generated summaries could be shown in podcast apps as additional information for each episode. In addition to these obvious use cases in search systems and podcast overviews, both systems can be utilized in the context of voice assistants, like Siri or Cortana. The user could ask the assistant to play podcast segments about certain topics or have the assistant give summaries of newly released episodes in the user's feed.

For the task of retrieving podcast snippets we focus on re-ranking the search results depending on the following criteria established by the shared task:

- whether the segment is entertaining to the listener
- whether the speakers in the segment express an opinion
- whether the segment contains discussion between multiple speakers

In this first task, we aim to to investigate whether the inclusion of audio features benefits the re-ranking process, compared to utilizing only text features. Therefore, we propose four solutions. The first solution acts as a simple baseline to compare the other proposals against. It utilizes the standard retrieval model BM25 [Robertson et al., 1994] without performing any re-ranking. The three subsequent approaches also use BM25, but re-rank the results in different ways. One approach incorporates only text features. Another approach employs only audio features. The last approach utilizes a combination of text and audio features.

For the task of summarizing podcast episodes we propose two approaches. We aim to produce summaries that are intended to entice listeners by giving preference to entertaining segments while generating the summaries. For this, both approaches resort to attributes that are inferred by a combination of

audio and text features. The first approach extracts segments from the episode that are deemed entertaining and feeds them into a state of the art BART summarization model [Lewis et al., 2020] to produce an abstractive summary. The second approach utilizes an adapted version of the TextRank algorithm [Mihalcea and Tarau, 2004] to produce an extractive summary. Based on both approaches, short audio clips intended to give an overview of an episode are also created.

Following this introduction, we present an overview of related work in the domain of retrieval and summarization for podcasts in chapter 2. Subsequently in chapter 3, we describe the task at hand and the methodological approach of finding suitable solutions as well as implementing them. Chapter 4 details the evaluation of our proposed approaches. A discussion about the evaluation results and the general results of this thesis is presented in chapter 5. Finally, we formulate a conclusion in chapter 6.

Chapter 2

Related Work

In this chapter we present an overview of approaches in the fields of retrieval and summarization of podcasts. Both of these fields are not widely studied yet, therefore the submissions to the first edition of the TREC podcast track in 2020 by Jones et al. [2020] constitute the main part of this chapter. The TREC podcast track features two shared tasks, in which approaches for podcast retrieval and summarization are submitted. Both shared tasks are based on the dataset "100,000 Podcasts: A Spoken English Document Corpus" published by Clifton et al. [2020], which consists of audio files, automatic transcriptions and metadata for more than 100,000 podcast episodes. An overview of the dataset is presented in section 3.2.

2.1 Retrieval

In a study conducted by Besser et al. [2008] about user goals and strategies in podcast search, the authors find that user goals in this field seem to be different from goals in general web search. Users, for a large part, search for podcasts for personal opinions or ideas, for detailed information about topics or as a combination of information and entertainment. Furthermore, the authors find that search strategies are highly influenced by the available tools, as most subjects expressed an interest in content-based search for podcasts, but did not see this as technically possible.

For the TREC 2020 podcast track retrieval task, text queries are supplied and the submitted retrieval systems are intended to produce a ranking of relevant podcast snippets. Snippets have a fixed length of two minutes and are overlapping by one minute. As described by Jones et al. [2020], many of the runs use pre-trained transfer learning models. Especially transformer architectures [Vaswani et al., 2017] are utilized in a large number of runs.

Additionally, a majority of runs utilize the more verbose topic description instead of the shorter topic query. In total, 24 retrieval runs were submitted by 7 participants. Only one run utilizes the supplied audio data. Submissions to the retrieval shared task are presented in the following.

Yu et al. [2020] submit a total of seven runs to the shared task, four of which are intended to be used as baselines. The first baseline run simply utilizes the BM25 retrieval function [Robertson et al., 1994]. The second baseline run uses Query Likelihood as its language model. The third baseline run retrieves results using BM25 and utilizes a BERT re-ranking model [Nogueira and Cho, 2019] pre-trained on MS MARCO passage retrieval data [Nguyen et al., 2016]. The supplied topic descriptions are used as the input for the re-ranking model. The fourth baseline is the same as the third baseline, except that the queries are used as the input for the re-ranking model instead of the topic descriptions. These four approaches also constitute the baselines for the general evaluation of the 2020 podcast track retrieval task by Jones et al. [2020]. Three more re-ranking approaches are also proposed. They also utilize a BERT model, but they are respectively finetuned on crowd-sourced data, synthetic data from generated questions and synthetic data from episode titles and descriptions. The approach using a BERT model finetuned on crowd-sourced data achieves the best results with a nDCG@20 of 0.473. However, when analyzing the nDCG with no cutoff, the baseline approach using BM25 scores the highest with a nDCG of 0.52.

Moriya and Jones [2020] submit a total of five runs using query expansion techniques. They extract nouns and named entities from the query description and add them to the query. Additionally, they deploy a pseudo-relevance approach that utilizes the first ten pages of search results of the Google Search API. Furthermore, an approach which utilizes pseudo-relevance feedback to derive hypernyms and hyponyms of query terms using WordNet[1] is introduced. Their five runs consist of different permutations of these three approaches. A combination of all three approaches achieves the best results with an nDCG with no cutoff of 0.586.

Sharma and Pandey [2020] submit a total of three runs using XLNet-based models [Yang et al., 2019] for document ranking. All approaches utilize BM25 and RM3 to retrieve the top 1,000 relevant snippets. These snippets are then re-ranked using different XLNet models finetuned on the MS MARCO Passage Ranking Dataset by Nguyen et al. [2016]. XLNet is chosen as it has no token limit for the input text, in contrast to models like BERT. The first approach uses a regression to return a score between 0 and 1. This score is then used for re-ranking. The second approach is a variant of the first and uses the last two

[1] `https://wordnet.princeton.edu`

hidden states of the model in the form of a concatenated vector as input to the linear layer. The third approach computes embeddings using the XLNet model for queries and documents. Cosine similarities between these embeddings are the main underlying basis of the re-ranking process. Their first approach using a regression achieves the best results with a nDCG with no cutoff of 0.5414.

Galuscáková et al. [2020] submit a total of five runs using combinations of multiple system variants. The first approach uses a sequential dependence model [Bendersky et al., 2010]. Retrieved documents are then re-ranked using a T5 model [Raffel et al., 2019] finetuned on the MS MARCO passage retrieval collection [Nguyen et al., 2016]. The second approach utilizes a sequential dependence model using concatenated title and description fields of the topics as queries; this approach is intended as a baseline. The third approach utilizes a combination of seven retrieval system with different language and relevance models. The results are combined and then re-ranked using the two transformer models BERT-Large and T5-Base. The fourth approach uses the same seven systems as the previous approach. However, the results of each system get re-ranked using only one of the two transformer models before combining them. The fifth approach is a combination of all previous four approaches. The third approach achieves the best results with a nDCG without cutoff of 0.6682.

2.2 Summarization

Klymenko et al. [2020] review the state-of-the-art of automatic summarization systems. As shown by the authors, current summarization systems are plagued by many deficiencies. Firstly, quality datasets for summarization are very rare. The most prominent example is the Daily Mail/CNN news dataset [See et al., 2017], but there exist barely any alternatives, which feature a big selection of texts with corresponding summaries. Additionally, they highlight a defining problem of the current state-of-the-art abstractive summarization systems, which are based on encoder-decoders, as they are mostly very limited regarding the maximum length of the input text.

Zheng et al. [2020a] present a baseline analysis for podcast abstractive summarization. Abstractive summarization is the generation of a summary in the form of newly formulated text. In contrast to extractive summarization, which extracts parts of the original text as a summary. According to the authors, existing abstractive summarization models are mostly trained on summarization datasets of professionally edited text, e.g. CNN/Daily Mail news. Podcasts however, are very different, as they are usually very lengthy, feature colloquial and conversational content and additional contents, such as commercials or

ad reads. This leads to a challenging summarization task according to the authors. They analyze multiple baseline and state-of-the-art approaches using the Spotify podcast dataset by Clifton et al. [2020]. As baselines the authors extract varying amounts of tokens, either from the start or the end of the transcript. They find that the first 100 tokens of the transcripts result in the highest scores. The deployed state-of-the-art models are BART, T5 and Prophet-Net which all use the first 512 or 1024 tokens of the transcript as the input. However, their performance is comparable to the baseline approaches, from which they conclude that further research in this area is still needed.

For the TREC 2020 podcast track summarization task, podcast episodes are selected and the submitted summarization systems are expected to produce a short text summary for each episode. As found by Jones et al. [2020], all submitted runs generate abstractive summaries, even though some runs utilize extractive techniques as intermediate steps. Furthermore, all runs utilize some form of deep learning model with the vast majority being based on a transformer architecture [Vaswani et al., 2017]. Interestingly, the 13 best performing approaches based on manual assessment all use some form of BART summarization model [Lewis et al., 2020]. Even the baselines using BART outperform the other baselines that do not use BART. In total, 22 summarization runs were submitted by 8 participants. No run utilizes the supplied audio data. Submissions to the summarization shared task are presented in the following.

Five baselines for the TREC podcast summarization track are proposed by Jones et al. [2020]. The first baseline only extracts the transcript of the first minute of the episode. Another baseline utilizes a BART summarization model that was trained on the Daily Mail/CNN news summarization corpus. Similarly, the same model was finetuned on a filtered version of the podcast corpus using the creator descriptions as summaries. Additionally, two extractive approaches are presented using the TextRank algorithm proposed by Mihalcea and Tarau [2004], which depicts text segments as nodes in a graph and utilizes a similar method to the PageRank algorithm to rank the text segments. In the first TextRank approach, the transcript gets divided into one minute segments and the most central segment is extracted using TextRank. The second TextRank approach segments the transcript into sentences using SpaCy[2] and extracts the two most central sentences. Manual assessment results show that the baseline using BART finetuned on the podcast descriptions scores the highest.

Owoicho and Dalton [2020] propose three summarization runs based on

[2] `https://spacy.io`

T5 summarization models [Raffel et al., 2019]. The first approach utilizes a T5 model finetuned on episode descriptions in the podcast dataset by Clifton et al. [2020]. The second approach utilizes the same T5 model, but generates the summary based only on the first 15 sentences in the transcript. The third approach also utilizes the same T5 model, but uses the 15 most important sentences as input. A SpanBERT model [Joshi et al., 2019] is used to extract the most important sentences. Manual assessment results show that the first approach scores the highest.

Manakul and Gales [2020] propose four summarization runs based on different BART summarization models. The first approach is intended as a baseline and utilizes a BART model finetuned on truncated podcast transcripts. For the other approaches, the authors train a hierarchical model [Manakul et al., 2020] on podcast transcripts without truncation to infer importance of sentences. The second approach uses a BART model finetuned on podcast transcriptions filtered using the hierarchical model. The third approach utilizes an ensemble of three BART models, all trained on podcast transcriptions filtered using the hierarchical model and other criteria. The fourth approach is the same as the third approach, but it utilizes nine instead of three BART models. Manual assessment results show that the third approach scores the highest.

Song et al. [2020] propose two summarization runs using different methods of selecting the input text for a BART summarization model. The BART model is firstly finetuned on the the Daily Mail/CNN corpus and secondly on the podcast corpus. The first approach simply uses the lead sentences in the transcript as input for the model. The second approach encodes each candidate sentence using a RoBERTa model [Liu et al., 2019] and selects important sentences based on multiple factors. These sentences are them used as input for the summarization model. Manual assessment results show that the second approach scores the highest. The authors find that it is very beneficial to identify important segments in the podcast transcript to use as the input for abstractive summarization models.

Rezapour et al. [2020] propose two abstractive summarization runs based on BART that take the genre of the podcast and named entities into account. As podcast vary widely in their format, the presented approaches generate summaries that are tailored to the style of the input podcast.

The length of the transcripts is a challenge for many teams, as it exceeds the maximum input length of abstractive summarization models in most cases. To combat this issue, multiple solutions are proposed. Karlbom and Clifton [2020] present a BART summarization model with the Longformer attention mechanism by Beltagy et al. [2020], which enables much longer input texts. To combine extractive and abstractive summarization techniques, Kashyapi and

Dietz [2020] extract the most salient segments of the transcript using an extractive model and use them as the input for an abstractive model. Furthermore, Zheng et al. [2020b] propose a two-phase-approach, in which they select important sentences from the transcript based on sentence similarity. Afterwards, they use these sentences as input for a pre-trained encoder-decoder summarization model. As an approach for extractive summarization of another form of long text, Miller [2019] leverages BERT sentence embeddings to generate summaries for lectures. K-Means Clustering is used to extract the sentences that are the nearest to the cluster centroid's. These sentences constitute the extractive summary.

Similiarly to automatic text summarization, a short audio clip can be automatically generated for a podcast episode. This is however a very sparsely studied field. Zhu [2021] presents various approaches to detect hotspots suitable to be included in an audio trailer. Speech emotion recognition, laughter detection and music detection are used in the selection process of these hotspots.

Featuring similar attributes as audio clips intended as trailers, automatic trailer generation for movies is explored by Irie et al. [2010], who extract segments with symbols (e.g. titles or names) and segments that feature impressive visual features or speech. Brachmann et al. [2009] as well as von Wenzlawowicz and Herzog [2012] analyze video and audio features to extract segments featuring specific attributes from a movie. These segments are then arranged according to specially designed sets of rules to form a trailer.

2.3 Audio

Berlage et al. [2020] explore the usage of audio embeddings for the task of topic segmentation of radio shows. They show a significant improvement of 32.3% compared to the F1-Measure of their text-only baseline.

Chapter 3

Methodological Approach

In this section, we present the process of finding suitable solutions for the problems at hand and implementing these solutions. For this, we firstly present a description of the task at hand in section 3.1. Secondly, we give an overview and analysis of the podcast dataset used in this thesis in section 3.2. Thirdly, in section 3.3 we present the process of feature engineering the classification of podcast segments regarding the three re-ranking criteria. Namely, these are the criteria "entertaining", "subjective" and "discussion". Classification is done in section 3.4 using text data, audio data, as well as a combination of both. Following the classification, we utilize the classification models in our approaches for the retrieval task in section 3.5 and the summarization task in section 3.6. We propose four retrieval runs to compare: a baseline, a run based on text features, a run based on audio features and a run based on a combination of text and audio features. All runs consist of three parts, one for each re-ranking criteria. We also propose two summarization approaches, one being abstractive and one being extractive. Both approaches integrate the classification of entertaining snippets, as they are intended to prioritize entertaining aspects. We also propose two approaches for the generation of short audio clips that are expected to give the listener a sense of what a podcast episode sounds like.

3.1 Task Description

We describe both tasks at hand to the necessary extent for the reader to follow the remainder of the thesis. This thesis proposes approaches for both shared tasks of the TREC 2021 podcast track organized by Karlgren et al. [2021]. Therefore, the following task descriptions are based on the requirements of the podcast track.

```
<topics>
  ...
  <topic>
    <num>96</num>
    <query>walkable city</query>
    <type>topical</type>
    <description>I would like to hear reports from various cities and
        neighbourhoods that tell me if they support a life style without cars
        for either residents or visitors. Both positive and negative reports
        are relevant, but need to give more detail than just a mention without
         explanation.</description>
  </topic>
  <topic>
    <num>97</num>
    <query>smuggling</query>
    <type>topical</type>
    <description>I want to hear stories about smuggling. General discussion
        about smuggling without reference to actual events are not relevant.</
        description>
  </topic>
  <topic>
    <num>98</num>
    <query>nobel prize laureates</query>
    <type>topical</type>
    <description>I want to hear about Nobel prize laureates. Biographies
        including both personal and professional life is relevant. The segment
         must name the laureate to be relevant.</description>
  </topic>
  <topic>
    <num>99</num>
    <query>samin nosrat</query>
    <type>known-item</type>
    <description>I heard that Chef Samin Nosrat makes a surprise appearance on
         an episode of The Cut and I want to find it.</description>
  </topic>
  ...
</topics>
```

Figure 3.1: A selection of retrieval topics used for the TREC podcast track retrieval task.

3.1.1 Segment Retrieval

The retrieval task is defined as the problem of finding relevant podcast segments based on search queries from a corpus of podcast episodes. Retrieval topics are provided in typical TREC topic format, consisting of a topic number, a short query, a topic type and a description of the query. Two topic types exist, "topical" and "known-item". Topical topics are intended to retrieve relevant segments about a topic, whereas known-item topics are intended to retrieve specific segments that are known to exist, but need to be located. 80% of the supplied topics are of the type "topical". A selection of four example topics is given in figure 3.1.

Retrieved podcast segments should have a fixed length of two minutes.

Each segment starts on the minute and segments overlap each other by one minute. For example, the first segments in an episode are 0.0s-119.9s, 60.0s-179.9s and 120.0s-239.9s etc. Overlapping segments are used because of sentences or phrases that might otherwise be split up by segment boundaries.

Furthermore, the task contains an aspect of re-ranking relevant segments. Each topic is expected to be submitted in the form of four ranked lists. One ranked list of relevant segments and three re-ranked lists of the same segments, but ranked based on different criteria. However, re-ranking is not relevant for the known-item topics. The re-ranking criteria are phrased exactly as follows by Karlgren et al. [2021]:

- **Entertaining:** The segment is topically relevant to the topic description AND the topic is presented in a way which the speakers intend to be amusing and entertaining to the listener, rather than informative or evaluative.
- **Subjective:** The segment is topically relevant to the topic description AND the speaker or speakers explicitly and clearly express a polar opinion about the query topic, so that the approval or disapproval of the speaker is evident in the segment.
- **Discussion:** The segment is topically relevant to the topic description AND includes more than one speaker participating with non-trivial topical contribution (e.g. mere grunts, expressions of agreement, or discourse management cues ("go on", "right", "well, I don't know..." etc.) are not sufficient).

This thesis utilizes only a simple baseline for the retrieval task and instead focuses on the re-ranking aspect.

3.1.2 Summarization

The summarization task is defined as the automatic creation of short text snippets containing the most important contents of podcast episodes. Summaries are intended to be significantly shorter than the transcript of the episode itself.

Audio Clips

Additionally, a short audio file of up to one minute in length is required for each summarized episode. The contents of the audio file are selected from the episode. These audio files are intended to provide insights about how a podcast sounds like.

```
"words": [
  { "startTime": "11.400s", "endTime": "11.700s", "word": "Hello", "speakerTag": 3 },
  { "startTime": "11.700s", "endTime": "12.200s", "word": "everyone", "speakerTag": 3 },
  { "startTime": "12.400s", "endTime": "12.600s", "word": "and", "speakerTag": 3 },
  { "startTime": "12.600s", "endTime": "12.900s", "word": "welcome", "speakerTag": 3 },
  { "startTime": "12.900s", "endTime": "13.300s", "word": "back", "speakerTag": 3 },
  { "startTime": "13.300s", "endTime": "13.500s", "word": "to", "speakerTag": 3 },
  { "startTime": "13.500s", "endTime": "14.100s", "word": "technology", "speakerTag": 3 },
  { "startTime": "14.100s", "endTime": "14.700s", "word": "Tuesday", "speakerTag": 3 },
  { "startTime": "14.800s", "endTime": "15.300s", "word": "with", "speakerTag": 3 },
  { "startTime": "15.300s", "endTime": "16.100s", "word": "ttts.", "speakerTag": 3 }
]
```


Figure 3.2: Example transcript of a sentence from the dataset, reading "Hello everyone and welcome back to technology Tuesday with ttts.". Features timestamps as well as an identification to distinguish speakers.

3.2 100,000 Podcasts: A Spoken English Document Corpus

To give the reader an overview of the utilized podcast data, we describe the dataset in this section. Additionally, we describe the process of preparing the dataset for our usage in our approaches. All work done in this thesis is based on the dataset "100,000 Podcasts: A Spoken English Document Corpus" published by Clifton et al. [2020]. The dataset[1] can be accessed for research purposes and is also used in both iterations of the TREC podcast track. In fact, it was initially created for usage in the first iteration of the TREC podcast track.

3.2.1 Composition

Consisting of more than 100,000 randomly sampled episodes and nearly 60,000 hours of audio, the dataset is sufficiently large for retrieval tasks. Each episode is represented by the audio file in ogg format, an automatically generated transcript in json format and some metadata. Therefore, adequate data for approaches based on transcripts as well as on audio data is available.

Figure 3.2 shows a truncated version of the word-level transcript of the first sentence of an example episode. Each word includes information about the start time, end time and an automatically inferred speaker tag. As can be seen in the last word of the sentence, punctuation is included. Additionally, even the abbreviation "ttts", which stands for the creator of the podcast "Trevors Traveling Tech Support" is detected.

[1] `http://podcastsdataset.byspotify.com`

field	used	description
show_uri	✓	Spotify unique podcast identifier (format: spotify:show:[show_filename_prefix])
show_name		name of the podcast (set by creator)
show_description		description of the podcast (set by creator)
publisher		publisher of the podcast, e.g., podcaster
language		spoken language, e.g. "en", "en-US", "en-GB"
rss_link		link to RSS feed
episode_uri	✓	Spotify unique podcast episode identifier (format: spotify:episode:[episode_filename_prefix])
episode_name		title of the episode (set by creator)
episode_description	✓	description of the episode (set by creator)
duration		duration of the episode in minutes
show_filename_prefix		unique show ID (also included in the show_uri)
episode_filename_prefix		unique episode ID (also included in the episode_uri)

Table 3.1: Available metadata in the podcast dataset for each episode and which of these is used in this thesis.

The supplied metadata for each episode consists of the components described in table 3.1. In this thesis, we utilize the supplied unique identifiers of podcast shows and episodes to identify episodes. Furthermore, the supplied episode descriptions are used for evaluation purposes in the summarization task. Additionally, information available through the RSS feeds was crawled for every show and is included in the corpus as xml files. These files contain some additional information to the metadata presented in table 3.1, such as as a flag for explicit language, category information and links to podcast artwork. Unfortunately, this information is not consistent across all shows. Not all xml files include the same fields and the format in which the information is stored in the nodes is not standardized for every field, e.g. the explicit flag, which is stored as "Yes" or "No", "True" or "False" and similar phrasings that express the same information.

3.2.2 Characteristics

As described by Clifton et al. [2020], the dataset consist of 105,360 randomly sampled podcast episodes with an average length of 33.8 minutes. Multiple episodes of the same shows can be included, with about 52% of shows being included more than once. All episodes were published between January 1, 2019 and March 1, 2020. Using the language metadata tag and a language identification algorithm[2] on the description, the dataset is aimed to include

[2] `https://pypi.org/project/langid/`

only English episodes. However, some non-English episodes were not detected this way and are still included. Furthermore, the authors employ a proprietary speech detection algorithm to filter out episodes containing less than 50% speech to remove episodes containing mostly music or white noise. Transcripts were automatically created using Google's Cloud Speech-to-Text API.[3] This approach not only provides a transcript with casing and punctuation, but also includes automatically inferred speaker diarization and timestamps for the beginning and end of each transcribed word. The authors state a sample word error rate of 18.1%, but they believe the transcripts are valuable if the high noise is considered. As seen in figure 3.3, results of the automatic speaker diarization show that the vast majority of episodes feature two speakers. This is especially relevant for the re-ranking task concerning discussion in episodes. However, the speaker diarization in the corpus is very noisy. A manually analyzed sample of 20 episodes by the authors had errors in 11 episodes for the number of speakers and in 4 episodes for segmentation of speakers.

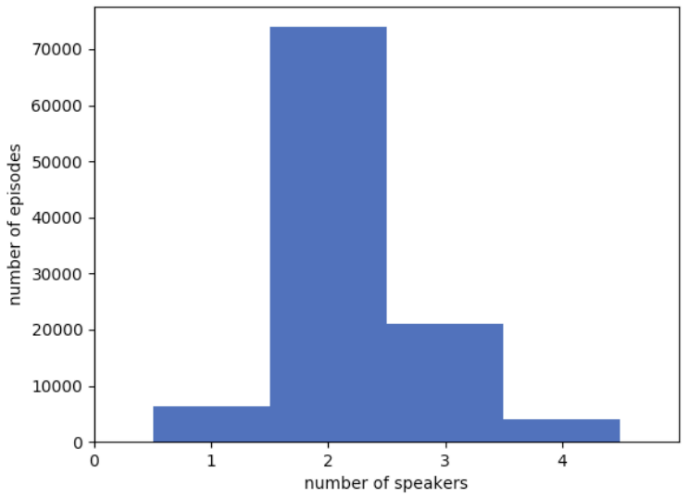


Figure 3.3: A visualization of the number of speakers per episode in the podcast dataset [Clifton et al., 2020].

The category of a podcast is set by the podcast creator and is included in the RSS data. When randomly sampling the episodes, category information was not taken into account. The supplied categories have to be used with caution according to Reddy et al. [2021], as they can be ambiguous. For example, the category "Kids & Family" includes both podcasts with content about the nurturing of kids and podcasts containing stories intended to be listened to by kids. Furthermore, Sharpe [2020] find that creators do not always

[3] `https://cloud.google.com/speech-to-text`

set the appropriate category for their podcasts. However, we can consider the category distribution of episodes in the dataset as a general overview. We aim to generate summaries that prioritize entertaining aspects in podcast episodes. The category of the podcast can therefore have an impact on our created summaries, as episodes from categories like "Comedy" generally feature more entertaining segments than categories like "News". Furthermore, the category of a podcast could potentially have implications on the title and descriptions of episodes, as informational podcasts may be more truthful about their contents, whereas comedic podcast may be inclined to utilize the titles and descriptions for comedic effect instead. For example, the episode "Aftermath (2020)"[4] of the history podcast "Throughline" features the following descriptive description:

> "In 1927, the most destructive river flood in U.S. history inundated seven states, displaced more than half a million people for months, and caused about $1 billion dollars in property damages. And like many national emergencies it exposed a stark question that the country still struggles to answer - what is the political calculus used to decide who bears the ultimate responsibility in a crisis, especially when it comes to the most vulnerable? This week, the Great Mississippi Flood of 1927 and what came after."

In contrast, the episode "Rob Almost Fights Some Guy Outside A Hamburger Store"[5] of the comedy podcast "The Always Sunny Podcast" features the following humorous, but very short and inexpressive description:

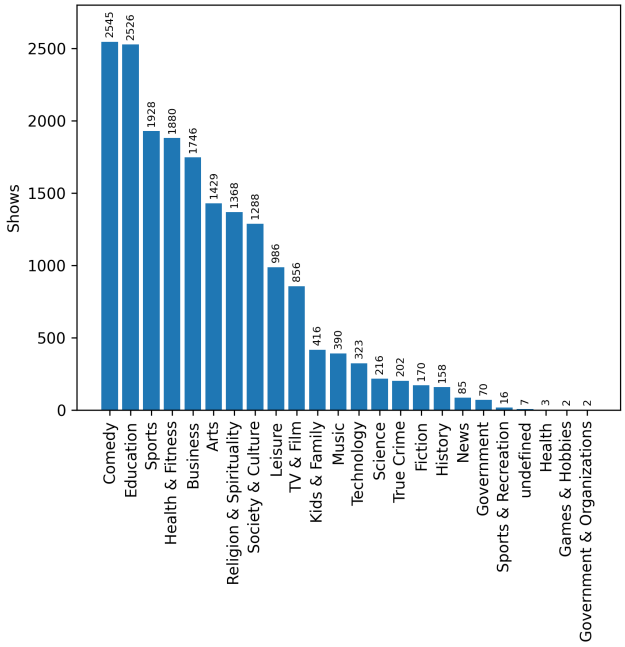
> "Everything is 100% fine."

Figure 3.4 shows the category distribution in the dataset. As visible in figure 3.4a, the most common category for podcast shows is "Comedy" with 2,545 shows. However, taking the number of episodes per show into account as seen in figure 3.4b, "Comedy" only ranks in sixth place with 10,580 episodes. Instead, "Education" is the most common category in the dataset.

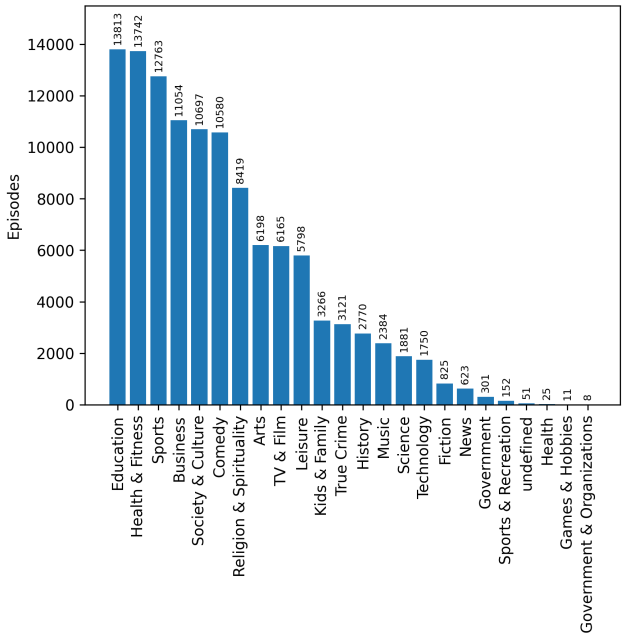
3.2.3 Dataset Preparation

To aid both the retrieval and the summarization task, we prepare the dataset in three different ways by indexing the episode transcripts in the form of chunks with lengths that are suitable for each task.

[4] `https://open.spotify.com/episode/5by5WUv6T0H818I7IQEadx`
[5] `https://open.spotify.com/episode/7K3h1Jw4LaoaTqSybqJnw9`



(a) Shows



(b) Episodes

Figure 3.4: A visualization of the distribution of categories for shows and episodes in the podcast dataset.

Segment Retrieval

All podcast episodes in the corpus are split into overlapping chunks of two minutes in length. Each chunk starts on the minute which leads to one minute of overlap, e.g. 0.0-119.9 seconds, 60.0-179.9 seconds, 120.0-239.9 seconds etc. We index these chunks using ElasticSearch.[6] Each document in the index contains a unique ID of the episode and a unique ID of the show. Additionally, start time, end time and length of the chunk in seconds are included. A transcript of the chunk and the included number of words are also available for each document. Inferred values for the three re-ranking criteria are also included. Each criterion is inferred in three ways (text, audio and a combination of both) and therefore included three times each. In total, about 3.4 million documents are indexed.

Summarization

We utilize two ElasticSearch indices for the summarization task, which features a reduced dataset of only 965 episodes.

Firstly, we create an index with sentence level segmentation of each transcript. We utilize the NLTK tokenizer with the Punkt sentence tokenization model.[7] Each sentence gets indexed as a separate document and includes a unique ID of the episode the sentence is contained in, as well as start time, end time and length of the sentence in seconds. Additionally, the position of the sentence in the episode, time stamps for the beginning and end, a transcription of the sentence and an inferred value for entertainment in this sentence are included in each document.

Secondly, another index contains the whole transcript of each episode. Each document contains a unique ID of the episode, the length of the episode in seconds, the number of sentences contained in the episode, as well as a full transcript.

3.3 Feature Engineering: Entertaining, Subjective, Discussion

In this section, we aim to classify segments of podcasts regarding three criteria based on the transcription, the audio content or a combination of both. The three criteria we consider are:

[6] `https://www.elastic.co/de/elasticsearch/`

[7] `https://www.nltk.org/api/nltk.tokenize.html`

- **Entertaining:** The segment is intended to entertain the listener rather than inform them.
- **Subjective:** The speakers in the segment express their opinions on a topic.
- **Discussion:** More than one speaker participate actively in the segment. Active participation is defined as non-trivial topical contribution.

As we aim to classify the three criteria in three different ways (using audio, using text and using both), we conclude with nine classification models. We also differentiate the occurrence of criteria in segments by calculating a confidence value, which is represented by a numerical value between 0.0 and 1.0. The trained models are utilized in both section 3.5 and section 3.6.

We do not have access to annotated data for the three criteria to train these models, therefore we annotate some data on our own. However, this is a very time consuming task and we require a sufficient amount of training data for typical supervised classification models. Therefore, we utilize transfer learning techniques for all approaches. We pre-train our own model for the audio classification and utilize an already pre-trained model for the text classification. These models are used to generate rich vector representations, further called embeddings, which we use as a base for the supervised training of our classification models.

3.3.1 COLA Training

With the aim of creating embeddings from audio clips, we train a model unsupervised on audio files from the podcast dataset. We utilize the COLA (contrastive learning for audio) approach proposed by Saeed et al. [2020]. According to the authors, this approach is suitable to tasks with limited training data as it outperforms even some supervised approaches. Additionally, table 3.2 shows the performance of COLA compared to other self-supervised methods for classification tasks on various audio datasets. Shown for comparison are standard triplet loss, two AUDIO2VEC (CBoW and SG) and temporal gap prediction models [Tagliasacchi et al., 2019, 2020], as well as a TRILL-19 system [Shor et al., 2020]. The authors show that COLA outperforms most of the other approaches.

COLA learns general purpose audio representations from unlabeled audio data (visualized in figure 3.5). To achieve this, a neural network is pre-trained on this data with a contrastive loss function. Multiple segments are extracted from audio clips and converted to log-compressed mel-filterbanks. At each step,

	CBoW	SG	TemporalGap	Triplet Loss	TRILL	COLA
speaker Id. (LBS)	99.0	**100.0**	97.0	**100.0**	-	**100.0**
speech commands (V2)	30.0	28.0	23.0	18.0	-	**62.4**
acoustic scenes	66.0	67.0	63.0	73.0	-	**94.1**
birdsong detection	71.0	69.0	71.0	73.0	-	**77.0**
music, speech and noise	98.0	98.0	97.0	97.0	-	**99.1**
music instrument	33.5	34.4	35.1	25.7	-	**63.4**
speech commands (V1)	-	-	-	-	**74.0**	71.7
speaker Id. (Voxceleb)	-	-	-	-	17.7	**29.9**
language Id.	-	-	-	-	**88.1**	71.3
average (TRILL tasks)	-	-	-	-	59.9	57.6
average (non-TRILL)	66.25	66.0	64.3	64.4	-	82.5

Table 3.2: Test accuracy (%) of a linear classifier trained on top of COLA embeddings or baseline pre-trained representations [Saeed et al., 2020].

one segment is used as an anchor. Segments from the same audio clip are positive classes and segments from different audio clips are negative classes. Using contrastive learning, the model is trained to maximize agreement between an anchor and positive classes and minimize agreement between an anchor and negative classes. Afterwards, the encoder can be combined with additional classification layers for downstream tasks.

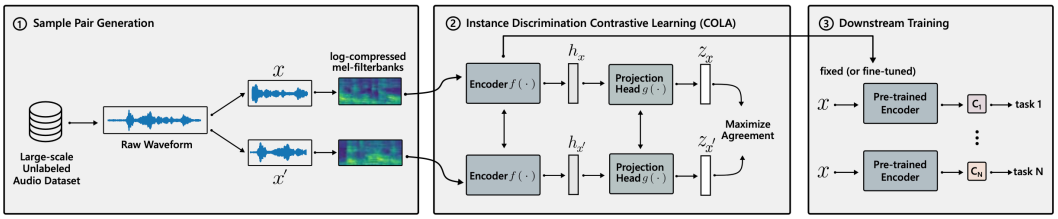


Figure 3.5: Overview of the contrastive self-supervised learning for audio [Saeed et al., 2020]. In this thesis, step 1 and 2 are executed on the audio data from the podcast dataset to train an encoder model. Step 3 is represented by the classification of podcast segments with models trained on manually annotated data.

For pre-training the COLA encoder model, we firstly randomly sample 10,000 hours of podcast episodes from the dataset using the Python random module.[8] For training, we utilize the COLA library[9] provided by Saeed et al. [2020], which uses TensorFlow.[10] To be able to use this library we adapt it with a custom data input pipeline for usage with our selected podcast episodes.

[8] `https://docs.python.org/3/library/random.html#random.sample`
[9] `https://github.com/google-research/google-research/tree/master/cola`
[10] `https://www.tensorflow.org`

# epochs	75					
learning rate	0.001					
embedding size	1,280					
batch size	32	64	128	256	512	1,024
time per epoch in seconds	2,200	3,000	4,300	8,900	17,500	44,700
time for 75 epochs in days	1.91	2.60	3.73	7.73	15.19	38.80

Table 3.3: Pre-training time of the COLA encoder model for increasing batch sizes. According to Saeed et al. [2020], larger batch sizes (up to 1,024) lead to better downstream performance.

Because of the large amount of data, which exceeds multiple terabytes, we convert our dataset to a TFRecordDataset[11] to improve input performance.

Regarding pre-training parameters, Saeed et al. [2020] find that, on average, a batch size of up to 1,024 provides better representations compared to smaller batch sizes. But increasing the batch size even more to 2,048 leads to a decline in performance. However, because of time limitations on the utilized compute cluster, we are unable to pre-train with a batch size this high. We therefore start the pre-training of multiple models with different increasing batch sizes, each on eight NVIDIA GeForce GTX 2080ti graphic cards. The time per epoch is recorded. All models are pre-trained with ADAM [Kingma and Ba, 2015] and a learning rate of 0.001 for 75 epochs. The embedding size is 1,280 for all. Table 3.3 shows the progression of training time per epoch and the calculated total time it would take to finish completely. Considering the limited computation time on the used cluster of a maximum of ten consecutive days, we choose the highest batch size, which is still able to be pre-trained fully for all 75 epochs. A batch size of 256 is therefore chosen and a COLA encoder model is fully pre-trained on the selected 10,000 hours of podcast audio data with this batch size.

3.3.2 Manual Annotation

We manually annotate 1,000 podcast segments (two minutes long each) to use as training data for our classifiers. For segment selection, we create a list of 100 queries. The queries consist of the train and test topic set of the segment retrieval task of the TREC 2020 podcast track [Jones et al., 2020], queries based on a dataset published by Kasturia et al. [2022] and a selection of newly created queries. A complete collection of all 100 queries is presented in appendix A. Each query is used as the input for a search in the ElasticSearch

[11] `https://www.tensorflow.org/api_docs/python/tf/data/TFRecordDataset`

podcast segment corpus using BM25. For each query, the 10 most relevant results are added to the annotation set.

We utilize the open source data labeling tool Label Studio[12] for our annotation process. A screenshot of the annotation process is shown in figure 3.6. The annotation process is based on the evaluation process proposed for the segment retrieval task of the TREC 2021 podcast track [Karlgren et al., 2021]. Each segment is presented with the corresponding audio clip in playable form. Annotators are instructed to listen to the whole segment. Additionally, annotators have access to the transcript of the segment, as well as the preceding and following transcripts for additional context. Furthermore, the query, the query type and a description are provided. Annotators are instructed to grade the segments on their relevance for the used query and on the three re-ranking criteria. Multiple choice answers are provided. Relevance is graded on a scale of Bad (0), Fair (1), Good (2) and Excellent (3). The re-ranking criteria are graded with three options. A segment is either adhering, partially adhering or non-adhering to the criteria. The annotation process is completed by four people. A sample of 15 segments is annotated collectively to ensure a uniform comprehension of the labels. Each remaining segment is annotated once. The total length of all segments is 33.33 hours.

Figure 3.7 shows the distribution of the annotations concerning the relevance of segments. "Bad" is the most common annotation. However, "Excellent" is ranked second. The average relevance is 1.61.

Figure 3.8a shows the distribution of the annotations concerning the adherence of the re-ranking criterion "entertaining". A majority of segments are classified as not entertaining. The average value is 0.696.

Figure 3.8b shows the distribution of the annotations concerning the adherence of the re-ranking criterion "subjective". A majority of segments are classified as subjective. The average value is 1.269.

Figure 3.8c shows the distribution of the annotations concerning the adherence of the re-ranking criterion "discussion". Very few segments are classified as partially adherent. The amount of segments containing discussion is only slightly higher than the amount of segments containing no discussion. The average value is 1.07.

3.3.3 Audio Classification

We aim to classify podcast segments for the three re-ranking criteria based on the audio data. Segments are supposed to be classified into two classes, they are either adherent or non-adherent. For this, we intend to train one classifier

[12] `https://labelstud.io`

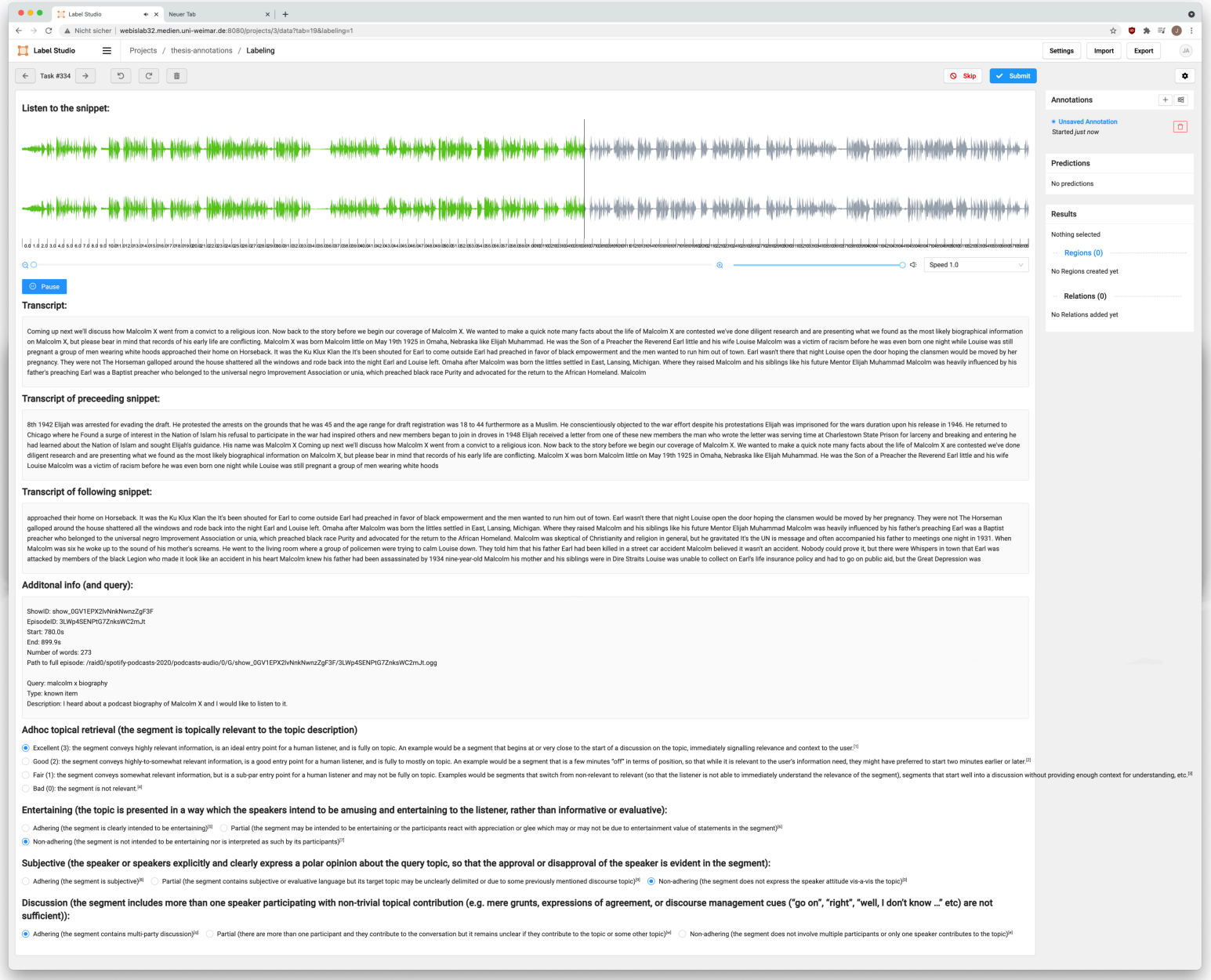


Figure 3.6: Interface used for the manual annotation process for the criteria "entertaining", "subjective" and "discussion" for podcast segments.

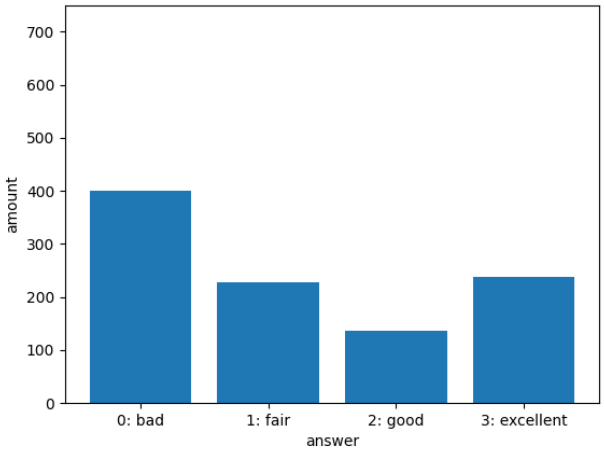
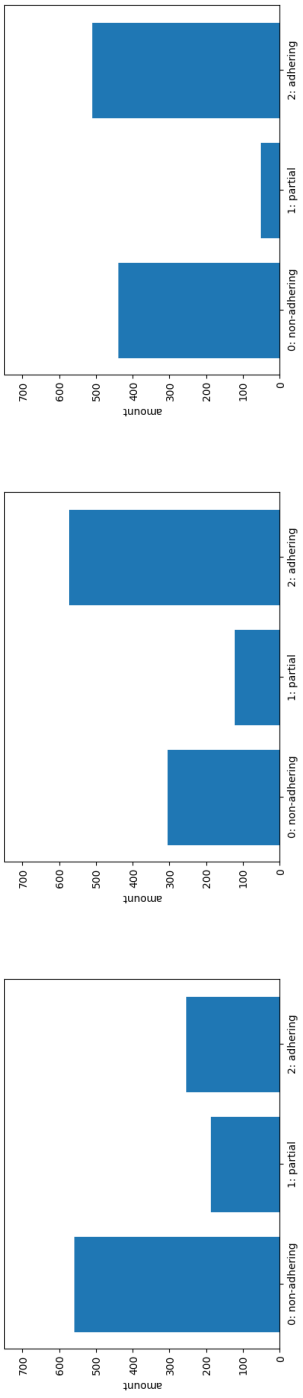


Figure 3.7: Distribution of 1,000 manual annotations regarding the relevance of the segments to the used queries.

for each criterion. However, we only have a very limited amount of annotated training data in our 1,000 manual annotations created in section 3.3.2. Therefore, we utilize the COLA model we trained in section 3.3.1 with the intention of leveraging the rich representations of audio data learned by the model. Utilizing the COLA model, we generate embeddings with a size of 1,280 for each of the 1,000 audio clips in our training set. Three classifiers for each criterion are trained with the training data on these embeddings. For this, we utilize the machine learning library scikit-learn [Pedregosa et al., 2011].[13] We choose a k-Nearest-Neighbors Classifier, a Random Forest Classifier and a Linear Support Vector Classifier. The k-Nearest-Neighbors Classifier utilizes k=5 neighbors, uniform weights and euclidian distance. The Linear Support Vector Classifier utilizes the penalization norm "l2", the loss function "squared-hinge", a regularization parameter C of 1.0, balanced class weight and a maximum of 1,000 iterations. The Random Forest Classifier utilizes 100 estimators, no maximum depth and a balanced class weight. As these are two class classifications, we map our annotations to two classes. Segments that are annotated as adhering for the re-ranking criteria are labelled as positives. In contrast, segments that are annotated as partially adhering or non-adhering are labelled as negatives.

In total, nine models are cross-validated in section 3.3.6 and one model is chosen for the classification of each re-ranking criterion.

[13] `https://scikit-learn.org/`



(a) Entertaining **(b)** Subjective **(c)** Discussion

Figure 3.8: Distribution of 1,000 manual annotations for the three re-ranking criteria.

3.3.4 Text Classification

We aim to classify podcast segments for the three re-ranking criteria based on the transcript data. However, because of the limited amount of training data, we utilize transfer learning in this approach as well. Similar to section 3.3.3, we generate embeddings on which we train our models. For the embedding generation, we utilize two pre-trained models, a BERT [Devlin et al., 2018] model[14] and a RoBERTa [Liu et al., 2019] model.[15] They are provided through the Hugging Face transformers library [Wolf et al., 2020].[16] For a fair comparison, we choose the same three classifiers with the same parameters as in section 3.3.3.

In total, 18 models are cross-validated in section 3.3.6 and one model is chosen for the classification of each re-ranking criteria.

3.3.5 Combined Classification

We aim to classify podcast segments for the three re-ranking criteria based on a combination of the transcript data and audio data. In this approach, we combine the approaches of sections 3.3.3 and 3.3.4. Therefore, we concatenate the audio embeddings and text embeddings und train the classifiers on these combined embeddings. Two versions are presented, a concatenation of COLA and BERT embeddings and a concatenation of COLA and RoBERTa embeddings. For a fair comparison, we choose the same three classifiers with the same parameters as in sections 3.3.3 and 3.3.4.

In total, 18 models are cross-validated in section 3.3.6 and one model is chosen for the classification of each re-ranking criteria.

3.3.6 Classification Model Selection

To select the best performing classification models, we perform a cross validation on all models. We utilize a stratified k-fold cross validation with $k = 10$. Therefore, the distribution of percentages for each class is preserved in all folds. A random state of 2021 is passed for reproducible output between different cross validations. The following metrics are used to score the cross validations:

- **Precision:** Precision is the fraction of positive samples correctly classified as positive among all samples that are classified as positive (either correctly or incorrectly). Larger values are better.

[14] `https://huggingface.co/bert-base-uncased`
[15] `https://huggingface.co/roberta-large`
[16] `https://huggingface.co/docs/transformers/index`

- **Recall:** Recall is the fraction of positive samples correctly classified as positive among all positive samples. Larger values are better.
- **F1-Score:** F1-Score measures the accuracy of a test by combining Precision and Recall in the form of a harmonic mean. Larger values are better.

We cross-validate all classifiers and compare the resulting scores. Figures 3.9, 3.10 and 3.11 show the distribution of scores in the form of boxplots for all five embedding types and all three classifiers. The edges of the boxes show the first and third quartile, as well as the median as a line in the middle. Whiskers represent minima and maxima and outliers are marked as dots. Figure 3.9 shows the scores for the re-ranking criterion "entertaining". Here, a wide distribution can be observed for almost all scores. Figure 3.10 shows the scores for the re-ranking criterion "subjective". Figure 3.11 shows the scores for the re-ranking criterion "discussion". Generally, the scores for the re-ranking criteria "entertaining" are worse than for the other criteria. The other two score similarly to each other.

For uniformity and fair comparison, we intend to select the same classifier for all three criteria. Judging the F1-Scores, the Linear Support Vector Classifiers score similarly to the other two for the re-ranking criteria "subjective" and "discussion". However, it achieves slightly better results for the criterion "entertaining". Therefore, we select the Linear Support Vector Classifier.

We also select the used embedding types for text classification and combined classification. The median F1-scores of RoBERTa scores are equal or higher than the BERT scores for all three criteria. For the text-based classification, we therefore select RoBERTa. Similarly, the combination of RoBERTa and COLA embeddings achieves higher F1-Scores than the combination of BERT and COLA for all three criteria. Therefore, select the combination of RoBERTa and COLA.

3.4 Classification

Based on the selection process described in the previous section, we train Linear Support Vector Classifiers for all three re-ranking criteria based on audio data (COLA), text data (RoBERTa) and the combination of both (COLA and RoBERTa). For the training, we utilize the complete annotated set of 1,000 podcast segments. We utilize these models to classify every document in the segment-wise index presented in section 3.2.3. However, instead of directly classifying the segments into two classes, we utilize the function "predict_proba(X)" provided by the Linear Vector Classifier impementation of

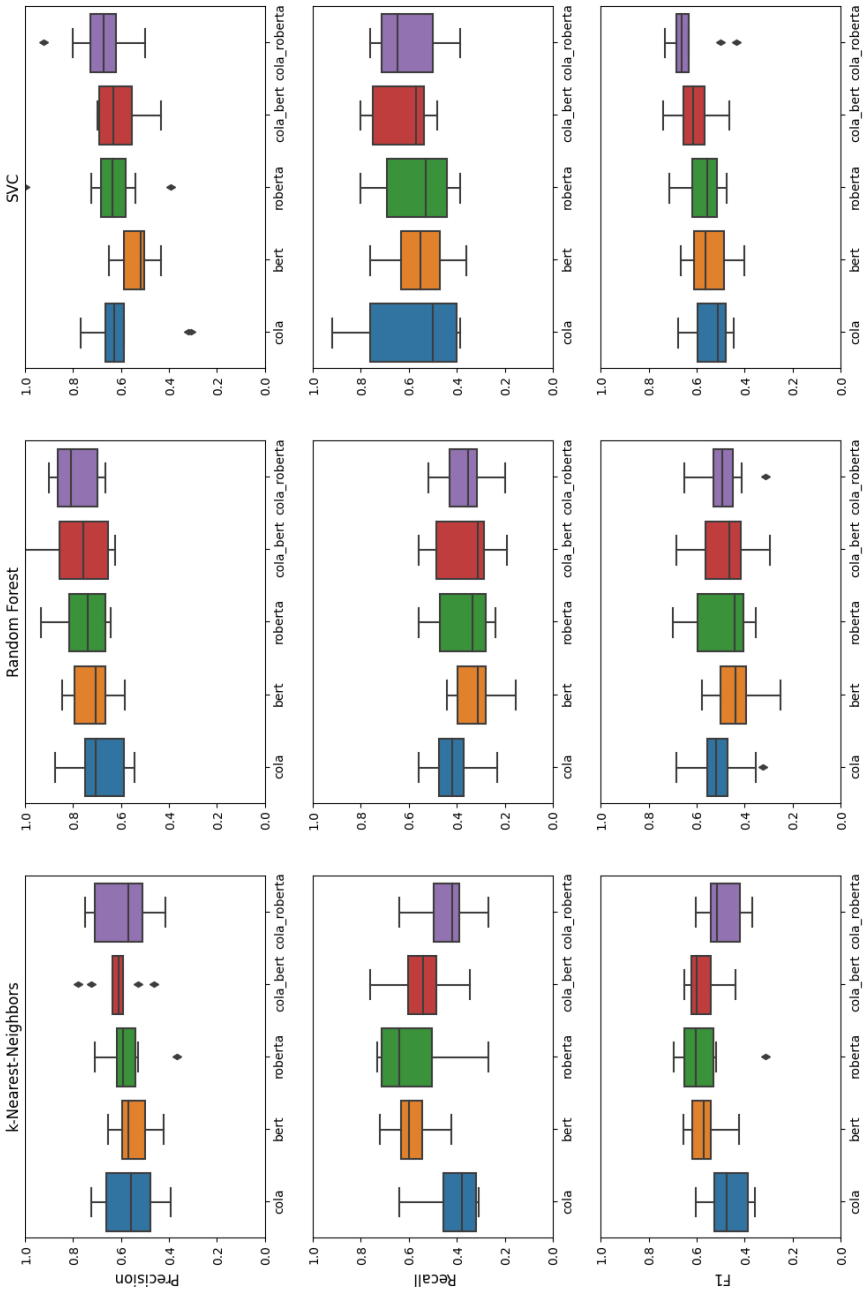


Figure 3.9: Results of cross validation for re-ranking criterion "entertaining". Shows classifiers in columns and metrics in rows.

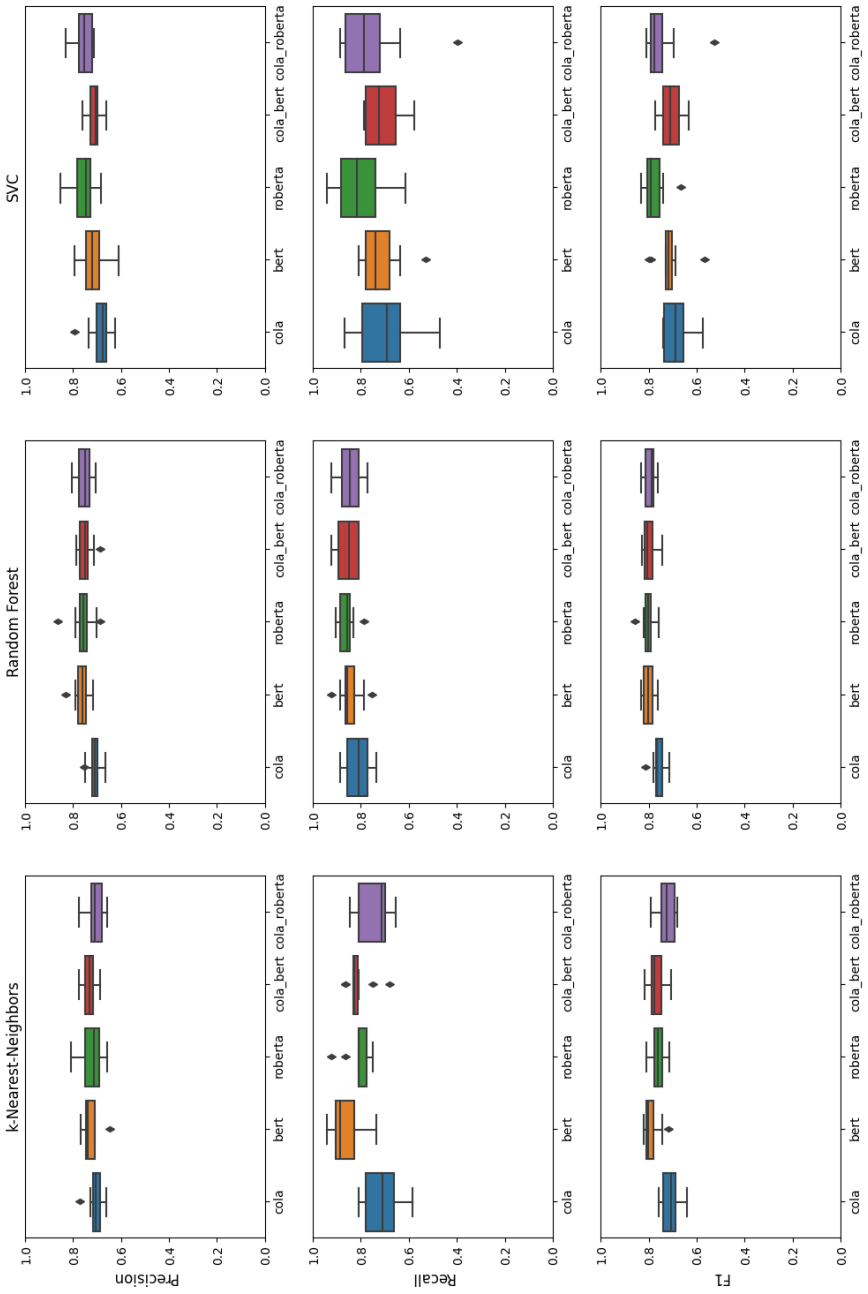


Figure 3.10: Results of cross validation for re-ranking criterion "subjective". Shows classifiers in columns and metrics in rows.

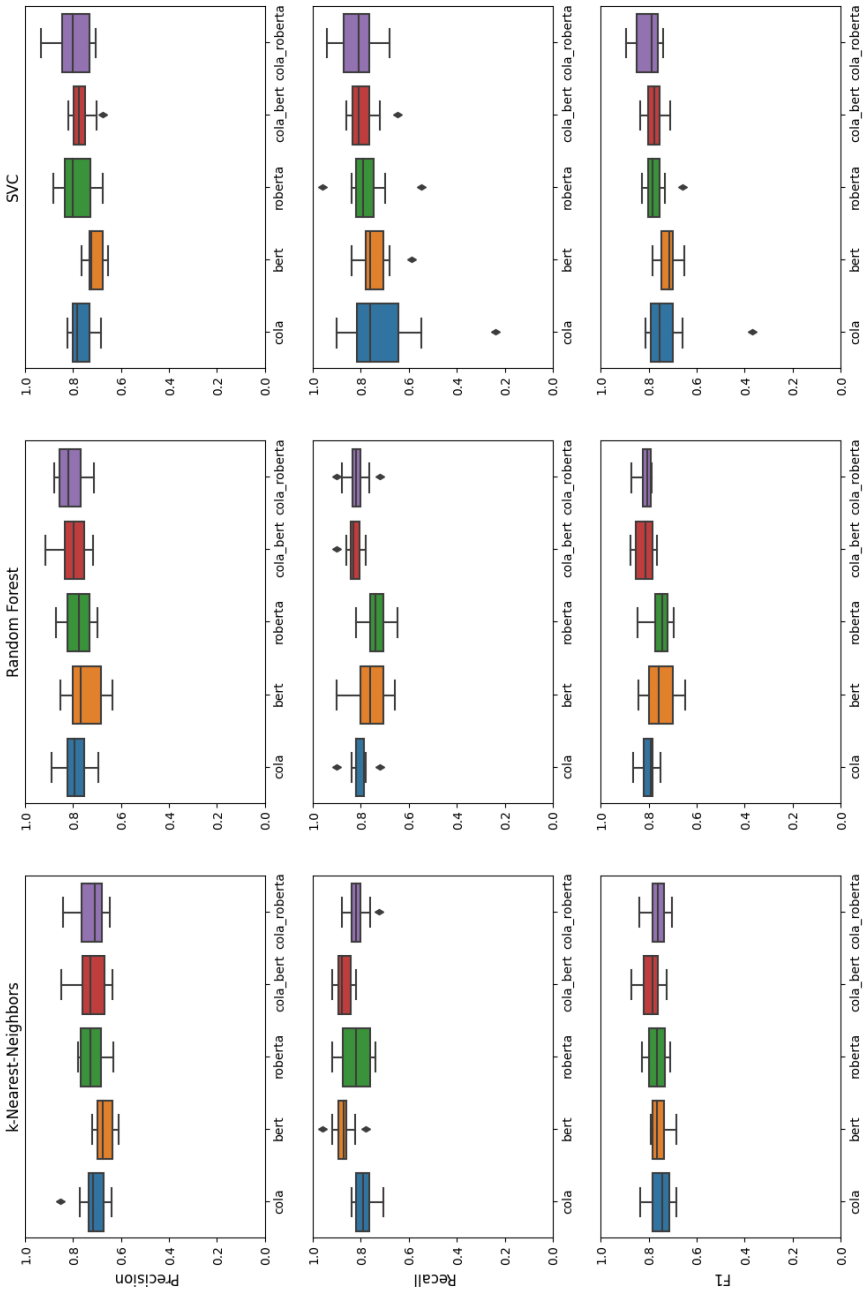


Figure 3.11: Results of cross validation for re-ranking criterion "discussion". Shows classifiers in columns and metrics in rows.

scikit-learn to generate probabilities for a positive classification. This means that a value between 0.0 and 1.0 is computed, which enables a better comparability between segments, in contrast to binary classes. Every segment in the corpus gets classified like this for all three re-ranking criteria, based on audio data, text data and the combination of both. Each segment is therefore classified nine times. Therefore, we add the following nine fields containing float values to every document in the index, where the first part represents the used embedding type (COLA, RoBERTa or COLA_RoBERTa) and the second part represents the re-ranking criterion:

```
cola_entertaining,
cola_subjective,
cola_discussion,
roberta_entertaining,
roberta_subjective,
roberta_discussion,
cola_roberta_entertaining,
cola_roberta_subjective,
cola_roberta_discussion
```

3.5 Segment Retrieval

We propose a total of four retrieval systems, one baseline and three systems that respectively re-rank based on audio data, text data and the combination of both. The three re-ranking systems are conceptualized almost identically. A string of arbitrary length is used as input and a ranked list of podcast segments is returned. If one of the three re-ranking criteria is selected, the ranked list gets re-ranked based on this criterion.

3.5.1 Baseline

The baseline is utilized as a system with no re-ranking at all for comparison against the other runs. It utilizes BM25 with ElasticSearch's default parameters of $k1 = 1.2$ and $b = 0.75$. When searching the index for a query, a maximum of 1,000 results are returned, sorted in descending order by their relevance score.

3.5.2 Re-Ranking Based On Audio Data

We re-rank the retrieved segments based on the selected re-ranking criterion using the features classified based on audio data. For the re-ranking process,

we firstly retrieve the top 1,000 results using the baseline (presented in section 3.5.1). Each document in this ranking contains a score. This score is a float value and represents the relevance of the document for the given query. A higher score signifies higher relevance. For this system, we utilize the re-ranking criterion classified using audio data. To retain the aspect of relevance in the ranking, we calculate new scores by multiplying the relevance score of each document in the ranking with the value for the corresponding re-ranking criterion (cola_entertaining, cola_subjective or cola_discussion) contained in the document. Therefore, a low value for the re-ranking criterion results in a high relevance penalty. In contrast, a segment with the same relevance score, but a higher re-ranking criterion value receives a lesser penalty and will be ranked higher in the end. At last, the list is sorted in descending order by the new score and returned as the new ranking.

3.5.3 Re-Ranking Based On Text Data

We re-rank the retrieved segments based on the selected re-ranking criterion using the features classified based on text data. This system functions almost the same way as the system described in section 3.5.2. The only difference is the usage of the criteria fields based text data instead of audio data. Therefore, the fields "roberta_entertaining", "roberta_subjective" and "roberta_discussion" are utilized.

3.5.4 Re-Ranking Based On Audio And Text Data

We re-rank the retrieved segments based on the selected re-ranking criterion using the features classified based on a combination of audio and text data. This system functions almost the same way as the system described in section 3.5.2. The only difference is the usage of the criteria fields based on audio and text data instead of only audio data. Therefore, the fields "cola_roberta_entertaining", "cola_roberta_subjective" and "cola_roberta_discussion" are utilized.

3.6 Summarization

We propose a total of two summarization systems, which both utilize the classification models proposed in section 3.4. To generate summaries that still highlight entertaining aspects of the podcast episodes, we utilize a model for the classification of the criterion "entertaining". As we intend to implement two systems, it is intuitive for one of them to generate an extractive summary and the other one to generate an abstractive summary.

The summarization systems receive the full transcript and the full audio file of a podcast episode as their input. They are intended to generate a short text summary that recollects the main contents of the episode, while being significantly shorter than the full transcript. Section 3.2.3 described the creation of two indices for this task. One index contains the whole transcript for each episode and the other index contains every sentence from each full transcript. With a simple retrieval operation, all sentences from a particular episode can be retrieved easily.

3.6.1 Entertainment Classification

Both approaches segment the full episode transcripts into sentences. We therefore, predict the value for the criterion "entertainment" for each sentence. For this, we utilize the Linear Support Vector Classifier presented in section 3.3.6. We choose the classifier using the combination of audio and text data (COLA and RoBERTa), because the median F1-Score, Precision and Recall are all higher than Linear Support Vector classifiers using the four other embedding types (see figure 3.9). With this classifier, we classify all sentences in the sentence-wise index the same way as in section 3.4. Therefore, every sentence in the index possesses an accompanying float value between 0.0 and 1.0, which represents the predicted entertainment value of the sentence.

3.6.2 Abstractive Summarization Approach

We summarize a podcast episode by creating an abstractive summary based on entertaining segments. For this, we need to select an abstractive summarization model. Additionally, we need to create a shortened input text, because the full transcript of an episode exceeds the maximum input length of most abstractive summarization models.

We select the five sentences that feature the highest classified value for the criterion "entertainment". To give these sentences additional context, we also select the two previous and the two following sentences each. Sentences at the start or the end of the transcript may have less context, because fewer than two sentences before or after them exist. Therefore, we select a total of at most 25 sentences. Some sentences may be included more than once because of overlap. The redundant sentences are removed. In the end, all selected sentences are concatenated to form one string by utilizing the order in which they occur in the original transcript. This string is used as the input for an abstractive summarization model, from which the model generates a short summary. For this, we choose a selection of three pre-trained abstractive summarization models. The selected models are:

- **BART**[17]**:** BART uses a Tranformer-based neural machine translation architecture and can be applied to a wide variety of text-based tasks [Lewis et al., 2020]. The utilized BART model is fine-tuned on the CNN / Daily Mail dataset [See et al., 2017] for summary generation. As described in section 2.2, the best performing summarization approaches in the TREC 2020 podcast track all utilize some form of BART model.

- **DistilBART**[18]**:** BART models are very large and usually feature very long inference times. Therefore, DistilBART is a distilled version of BART with a smaller size and shorter inference times [Shleifer and Rush, 2020]. It is fine-tuned on the CNN / Daily Mail dataset for summary generation [See et al., 2017].

- **PEGASUS**[19]**:** The PEGASUS model utilizes a Transformer-based encoder-decoder architecture [Zhang et al., 2020]. It is trained on both the C4 [Raffel et al., 2019] and the HugeNews datasets (includes XSum [Narayan et al., 2018] and CNN / Daily Mail [See et al., 2017] datasets) for summary generation.

For the selection of the abstractive model, we undertake an automatic evaluation to compare their performance and select one model for further use. In the context of the TREC 2020 podcast track [Jones et al., 2020], a set of 150 podcast episodes with summaries was provided. Six types of summaries are included for each podcast episode. Apart from the names of the summary types, no other information is given about them. The included summary types are:

- A summary created using Latent Semantic Analysis [Steinberger and Jezek, 2004].

- A LexRank summary [Erkan and Radev, 2004].

- A TextRank summary [Mihalcea and Tarau, 2004].

- A quasi-supervised summary.

- A supervised summary.

- The episode description set by the creator of the podcast.

Additionally, all included summaries are graded by manual assessments. We select the highest rated summary for each episode and utilizes them as a reference to evaluate our own systems against. For evaluation purposes we utilize

[17] `https://huggingface.co/facebook/bart-large-cnn`
[18] `https://huggingface.co/sshleifer/distilbart-cnn-12-6`
[19] `https://huggingface.co/google/pegasus-xsum`

		BART	DistilBART	PEGASUS
Rouge1	Precision	0.291	0.232	0.26
	Recall	0.312	0.260	0.264
	F1-Score	0.294	0.231	0.253
RougeL	Precision	0.197	0.152	0.166
	Recall	0.193	0.173	0.161
	F1-Score	0.191	0.152	0.163

Table 3.4: Average Precision, Recall and F1-Score for Rouge1 and RougeL for all three abstractive summarization models on 150 test episodes.

Rouge1 and RougeL scores calculated using the rouge-score Python module.[20] We generate summaries using all three abstractive models for all 150 episodes and compute the Precision, Recall and F1-Score of both Rouge scores. Average values are calculated of each metric for all 150 episodes. The resulting values are presented in table 3.4. The BART model achieves the highest average scores in all metrics.

However, because of limited local computational resources, the creation of summaries in the needed quantities required for the TREC summarization task is not possible using the BART or the PEGASUS model. Therefore, we select the DistilBART model for all further usage, despite its slightly worse performance in our evaluation.

3.6.3 Extractive Summarization Approach

We summarize a podcast episode by creating an extractive summary prioritizing entertaining segments. To generate an extractive summary, we utilize a standard extractive summarization approach: TextRank [Mihalcea and Tarau, 2004]. However, we adjust the approach by giving preference to entertaining sentences based on the classified value for the criterion "entertaining" by the previously proposed model.

We create a graph where each sentence in the podcast episode transcript represents a node. Each edge in the graph gets assigned an initial starting weight. This weight is the semantic similarity between the two sentences connected by the edge. We extract the semantic similarity by embedding the sentences using an XML-RoBERTa [Conneau et al., 2020] model[21] and calculating the cosine similarity of the vectors. Therefore, each edge in the graph is

[20] `https://pypi.org/project/rouge-score/`

[21] `https://huggingface.co/sentence-transformers/paraphrase-xlm-r-multilingual-v1`

		multiply_average	sum
Rouge1	Precision	0.123	0.134
	Recall	0.461	0.468
	F1-Score	0.181	0.195
RougeL	Precision	0.076	0.084
	Recall	0.293	0.299
	F1-Score	0.112	0.123

Table 3.5: Average Precision, Recall and F1-Score for Rouge1 and RougeL for the two weight adjustment variants on 150 test episodes.

represented by a similarity between 0.0 and 1.0. We intend to adjust all starting weights in the graph using the classified values for criterion "entertaining" (further called "entertainment value"). For this adjustment, we explore two variants for adjusting the weight between two sentences in the graph:

1. **multiply_average:** This variant calculates the average entertainment value of both connected sentences. The average value gets multiplied with the cosine similarity.

2. **sum:** This variant sums up the entertainment value of both connected sentences and the cosine similarity.

We evaluate both variants on the 150 manually summarized episodes, exactly as in section 3.6.2. Table 3.5 shows the average Precision, Recall and F1-Score for Rouge1 and RougeL. Variant 2 (sum) achieves a higher average Precision, Recall as well as F1-Score for both Rouge scores than variant 1 (multiply_average). Therefore, we select the second variant (sum).

We execute the PageRank algorithm on the graph using the networkx Python module.[22] For this, we utilize a damping factor of 0.85 and a limit of 100 iterations. Afterwards, we have a ranking of all sentences by relevance. We select the ten most relevant sentences. These ten sentences are then concatenated based on the order in which they appear in the complete transcript of the episode. These concatenated sentences constitute the extractive summary.

3.6.4 Audio Clips

Additionally, this thesis proposes the generation of short audio clips for podcast episodes. The audio clips can consist of one coherent segment of the episode

[22] `https://networkx.org`

or they can be assembled from multiple clips. Audio clips have a maximum length of one minute. We propose two approaches for this as well, both are directly based on the two proposed summarization approaches. As we intend to persuade potential listeners with these clips, both approaches prioritize entertaining segments of the episodes.

Approach 1: Anchor Points

This approach is based on the abstractive summary generation proposed in section 3.6.2. We select the five sentences as anchor points that feature the highest classified value for criterion "entertaining". To give these sentences additional context, we also select the two previous and the two following sentences. Sentences at the start or the end of the transcript may have less context, because fewer than two sentences before or after them exist. Therefore, we select a total of at most 25 sentences. Some sentences may be included more than once because of overlap. The redundant sentences are removed. In the end, all selected sentences are sorted by the order in which they occur in the original transcript.

All indexed sentences contain time stamps of their beginning and end. We utilize these time stamps to concatenate the audio of the selected sentences into a new audio file using FFmpeg.[23] If the complete audio clip exceeds a length of one minute, it gets trimmed after the end of the last sentence before the one minute mark.

Approach 2: TextRank

This approach is based on the extractive summary generation proposed in section 3.6.3. An extractive summary is generated exactly as previously proposed by executing the TextRank algorithm. The corresponding audio of the sentences in this summary are used to create the audio clip.

All indexed sentences contain time stamps of their beginning and end. We utilize these time stamps to concatenate the audio of the selected sentences using FFmpeg. If the complete audio clip exceeds a length of one minute, it gets trimmed after the end of the last sentence before the one minute mark.

[23] `https://ffmpeg.org`

Chapter 4

Evaluation

In this chapter, we evaluate the proposed approaches of this thesis. The classification of the re-ranking criteria is evaluated in section 4.1. Section 4.2 presents the evaluation of the segment retrieval approaches, in particular the aspect of re-ranking. An evaluation of both summarization approaches is shown in section 4.3. Finally, we present the evaluation of the two approaches for the creation of short audio clips in section 4.4.

Parts of this evaluation are results of the manual evaluation of the TREC 2021 podcast track organized by Karlgren et al. [2021], for which we submitted a total of four retrieval runs, two summarization runs and two runs for the creation of short audio clips about podcast episodes. Results from the manual evaluations by NIST assessors are described in individual subsections called "TREC Manual Assessment".

4.1 Criteria Classification: Audio, Text And Combined

We evaluate the classification of podcast segments for the three criteria "entertaining", "subjective" and "discussion" based on the three proposed techniques: utilizing audio data (COLA), utilizing text data (RoBERTa) and utilizing a combination of both types of data (COLA + RoBERTa). Firstly, we show an evaluation on the binary classification into classes that are either adhering or non-adhering to the criteria. Secondly, we present a selection of example segments with the corresponding classifications on a scale from 0.0 to 1.0 and compare them to the TREC assessments.

Figure 4.1 shows the classification results of predictions from the previously executed 10-fold cross validations (described in section 3.3.6) for all three criteria and all three proposed techniques. The results are presented as confusion

matrices. Class "0" denotes a segment non-adhering to the criteria, further called negative class. Class "1" represents adherence to the criteria, further called positive class. Resulting values from each fold for each class are aggregated as a sum. A perfect distribution of predictions would feature values only in the quadrants for true positives (lower right quadrants) and true negatives (upper left quadrants).

Figure 4.1a shows the results for the criterion "entertaining". The combined approach performs better than the other two. Generally, it can be seen that the identification of the negative class performs well for all three approaches, whereas the classification of the positive class leads to a higher proportion of errors. However, it needs to be noted that this criterion features a very uneven class distribution with the vast majority of labels being negative, as already presented in figure 3.8a.

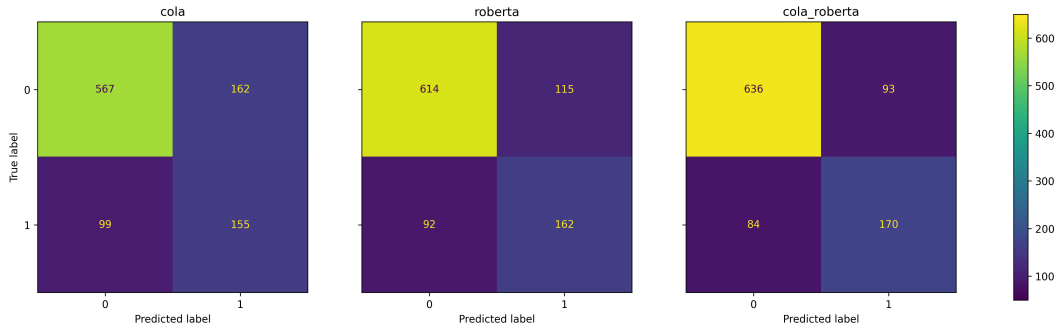
Figure 4.1b shows the results for the criterion "subjective". All three approaches perform very similarly. However, the approach utilizing text data performs slightly better than the other two when classifying the positive class, as it results in the least false negatives. The best performance of the classification of the negative class is achieved by the combined approach.

Figure 4.1c shows the results for the criterion "discussion". All three approaches perform similarly, with a slight overall advantage for the combined approach. However, the approach utilizing audio data results in the least amount of false negatives.

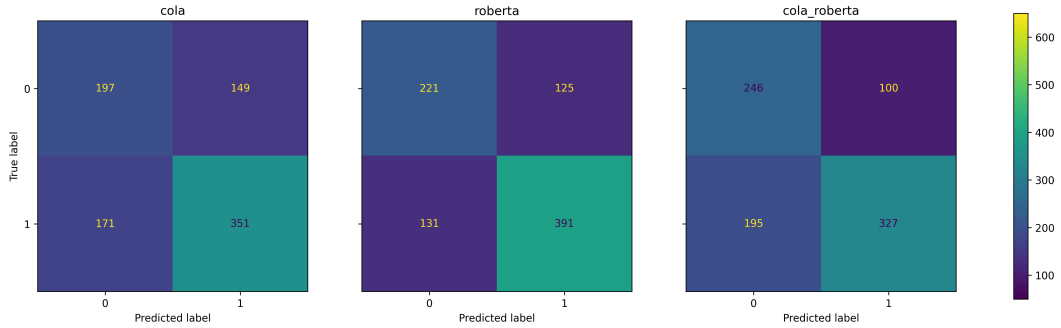
All in all, the combined approach achieves the best results. The combined approach performs the best for the criteria "entertaining" and "discussion". Additionally, for criterion "subjective" the combined approach performs comparatively to the other two. Generally, classification of the criterion "discussion" seems to perform better than the other two criteria.

4.1.1 TREC Manual Assessment

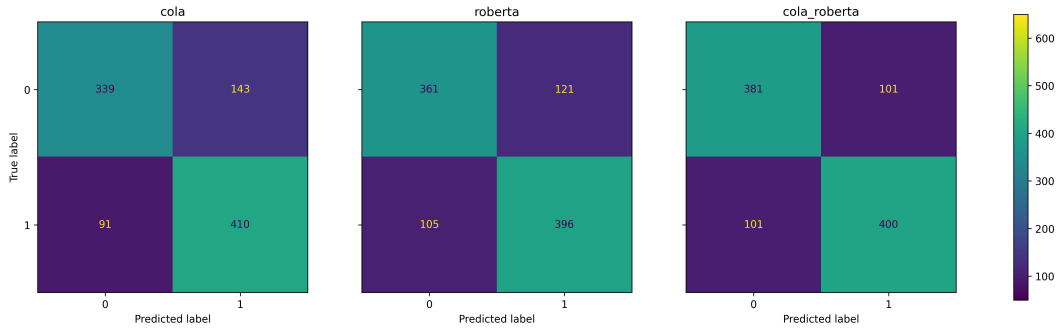
Furthermore, we present three podcast segments and their corresponding classifications, as well as the manual assessments of the TREC 2021 podcast track. We select segments that are classified particularly well or particularly bad for certain criteria or by certain approaches. Table 4.1 shows an overview of all values. Values for automatic classifications range from 0.0 (non-adhering) to 1.0 (adhering). Values for TREC manual assessment are 0 (bad), 1 (fair), 2 (good) and 3 (excellent). To enable comparison, we consider classifications < 0.5 and manual assessments of 0 and 1 as non-adhering. Classifications ≥ 0.5 and manual assessments of 2 and 3 are considered adhering. We show the transcript of the segment as it is contained in the corpus with no corrections or omissions. As the transcripts were created automatically, they are



(a) Entertaining



(b) Subjective



(c) Discussion

Figure 4.1: Confusion matrices for feature classifications using Linear Support Classifiers. Shows the three different approaches using audio data, text data and a combination of both. Calculated using a 10-fold cross validation by summing the predictions of each fold.

segment	criterion	audio	text	combined	TREC manual assessment
A	**entertaining**	0.28	0.66	0.61	2
	subjective	0.53	0.98	0.92	2
	discussion	0.37	0.95	0.94	2
B	**entertaining**	0.08	0.02	0.02	0
	subjective	0.71	0.35	0.61	0
	discussion	0.35	0.98	0.2	0
C	**entertaining**	0.29	0.04	0.09	0
	subjective	0.66	0.94	0.85	0
	discussion	0.65	0.54	0.64	2

Table 4.1: Classifications and manual assessments for all three presented segments. Values for automatic classifications range from 0.0 (non-adhering) to 1.0 (adhering). Values for TREC manual assessment are 0 (bad), 1 (fair), 2 (good) and 3 (excellent).

not 100% correct and contain errors. If a segment contains multiple speakers, speech of one speaker is distinguished in the transcript by an underline. This identification is done manually and not based on the speaker diarization in the corpus. No segments with more than two speakers are presented.

Segment A

We present the first podcast segment[1] from the podcast "Scammed". The episode is titled "15. Is Art a Scam?" and was released in December 2019. Two speakers are featured. The transcript reads as follows:

> "for a hundred twenty or $130,000 would have gone away. But the fact that this man ate it in public on video. Yeah and the gallery and the artist and the fucking buyer of the art is all fine with it. Yeah is what makes this so stunning so stun and the scammer scammed but helped prove that art is a scam. Yeah. It's just this is the best. It's like you have a feeling. Yeah. I see reading about it. And I think again I came back to that impulsive like looking at Modern Art and thinking what the fuck is that I can do that but truly like anytime I think so much of what you and I talked about is like around Financial scams. And where's the money going and who is scamming us? And who is who has all the money and where's the money going into the pockets? I am not mad at artist know I think all art is, you know, it's a conversation and it's a cultural Touchstone and I When people to be able to express themselves

[1] `https://open.spotify.com/episode/5fyJk5x020hTJgw96N1klK` (26:00-28:00)

> in any way that they can I think in in a vacuum all art should be free. But if you're a fucking rich-ass billionaire and you're going to spend $150,000 on a banana, yeah, like we should collectively be mad at the fact that that is allowed to happen. Yes, and that you don't have health insurance. Yes, like something else that I read. I'm like some fancy art blog today about this whole thing was like this is actually the fact that this sold liked. It's such a high price. Is another indicator that the economy is doing really well and that it's like obviously art and like, you know, non-essential things like go up when the economy is doing well and people have extra money to spend and I think even just the idea that it's sold three times that one thing like what else that our bottles sold three times. You can't I mean, I don't know I'm sure bigger artist did sell things but we don't know because it didn't splash the headlines and we don't care because nothing had an emotional reaction. Anyone, I guess what I'm saying is so many things that are that are probably in galleries or it's like one of a kind, you know what I mean? So it's I"

According to the manual assessment, the segment is adhering to all three criteria, as they are all assessed as 2. The approach utilizing text data and the combined approach classify accordingly, classifying all three criteria as adhering. Especially "subjective" and "discussion" are classified highly as > 0.9 by both. However, the approach utilizing audio data falsely classifies the criteria "entertaining" and "discussion" as non-adhering.

Segment B

We present the second podcast segment[2] from the podcast "Narcissism Recovery Podcast". The episode is titled "Personality Disorders from a Childhood Wounds Perspective" and was released in September 2019. One speaker is featured. The transcript reads as follows:

> "family of origin will be they By the way, then interact with the world because they have a distorted sense of self a damaged self and they will then see the world through this damage self and ultimately manifest very dysfunctional and chaotic relationships because of which the personality is largely built to protect this wound. It is not aligned with the authentic self. So specifically with narcissistic personality disorder. The individual is quite literally built in a completely entirely fabricated, excuse me sense of

[2] `https://open.spotify.com/episode/3OBHWHRK3TgF5JW21JhWpw` (4:00-6:00)

> self in order to Adopt that and protect from facing the true and battered sense of self. Now they are also looking to protect the the true self which is already wounded but often times after a while. They just quite literally adopt the false self as it is as if it is the only self that exists. Borderline personality disordered disordered individuals for example, the the personality is built around the abandonment wound and protecting from beginning of getting abandoned. So what happens is you have a child who's feels extremely fearful of getting thrown out of the house through abused and ultimately abandoned either physically or emotionally and therefore will create this very fragile needy sense of Personality where they constantly trying to maybe Cling to other people in order to protect themselves from getting abandoned here. We have the personality to quite literally developed around the fear of Abandonment in the case of NPD. It's built around feeling shamed and humiliated and to protect from further shame. So both maladaptive and ultimately the NPD protects kind of runs away from themselves into the false self to protect from facing the shame the humiliation of the true self. And ultimately not having to face that pain ever again a healthy personality should be an expression of the self and the mechanism used to communicate with the outside"

According to the manual assessment, the segment is adhering to none of the three criteria, as they are all assessed as 0. The criterion "entertaining" is classified correctly by all three approaches as < 0.1. However, the other two criteria vary widely for all three approaches. Especially "discussion" classified with the approach utilizing text data features a very high value of 0.98, even though there exists no discussion in the segment.

Segment C

We present the third podcast segment[3] from the podcast "BitcoinMeister- Bitcoin, Cryptocurrency, Altcoins". The episode is titled "This week in Bitcoin- 7-19-2019- Libra & BTC thoughts in the Philippines, Brazil, & South Africa!" and was released in July 2019. Two speakers are featured. The transcript reads as follows:

[3] `https://open.spotify.com/episode/2rbhmjbEMzgBTRrKPfhuP7` (28:15-30:15)
Note: The timestamps of the episode on Spotify and the given timestamps in the transcript are not exactly the same (in the transcript: 29:00-31:00). This could be explained by dynamically inserted advertisements [Mark, 2021], where the advertisements in already published episodes can be replaced. As newly inserted advertisement of a different duration than the old one would have an effect on the timestamps.

> "even some people on the other agencies our SEC the central bank. We have many people will they Pro crypto or Pro Innovation is tense. So I'd say it's never been better to work out regulations without with the current Administration in Brazil. All right. I've got some questions about the Bitcoin culture. There is a bunch of people trying to trade like a bunch of young people trying to flip. Our people mining are altcoins big. There are people using it for remittances What's the culture? Like I'd say it's mostly trading than I'd say using for Remington not really remittances, but International transfers, so paying for imports and trying to evade some of the the capital complete the import text that we have and some Financial taxes that we have over finding. Oh, Or International transfers, so that's how I would say most people are using but that's one thing is I think it's really plaguing the market for over five or six years, which are all of these scams pyramid scams Ponzi schemes using crypto using not using but just saying that they're investing or trading with Bitcoin or crypto and it's just I'll try to Ponzi schemes. We are still being plagued by this scams. well well and has the government tried to step in on that they are they are and that's the the challenge because sometimes the the ones that don't have a lot of information they will try to to pound it and try to step it on everything and try and thinking everything is the same and know you have Bitcoin is a true acids genuine acid digital asset budget also have it schemes and scams people trying to Take advantage of others. Are"

According to the manual assessment, the segment is only adhering to the criterion "discussion". "Entertaining" gets classified correctly by all three approaches. However, the criterion "subjective" is classified incorrectly by all three approaches. Especially the approach utilizing text data classifies the segment as > 0.9 for "subjective", even though it is manually assessed as non-adhering. "Discussion" gets classified correctly by all three approaches, but with values only slightly above 0.5.

4.2 Segment Retrieval: Re-Ranking

We evaluate the approaches for the task of re-ranking retrieved podcast segments based on the three re-ranking criteria. For this, we utilize the results of the manual assessments of the TREC podcast track.

re-ranking criterion	run	nDCG@30	nDCG@1,000	precision@10
entertaining	baseline	0.1182	0.2330	0.0975
	audio	0.0522	0.1748	0.0450
	text	0.0351	0.1584	0.0275
	combined	0.0332	0.1620	0.0275
subjective	baseline	0.1725	0.3435	0.2000
	audio	0.0591	0.2443	0.0600
	text	0.0371	0.2250	0.0350
	combined	0.0430	0.2320	0.0550
discussion	baseline	0.1619	0.3208	0.1600
	audio	0.0598	0.2289	0.0625
	text	0.0399	0.2101	0.0400
	combined	0.0475	0.2193	0.0550

Table 4.2: Re-ranking results of the manual assessment from the TREC 2021 podcast track. Shows all three re-ranking criteria and all three techniques (audio, text and combined). Additionally, a baseline without any re-ranking is presented.

4.2.1 TREC Manual Assessment

Table 4.2 shows the results of the manual assessment of the TREC 2021 segment retrieval task. Shown are the three approaches: utilizing audio data, utilizing text data and utilizing a combination of both. Additionally, results of a baseline approach utilizing BM25 without any re-ranking are presented for comparison. Presented evaluation metrics are nDCG with a cutoff at 30, nDCG with a cutoff at 1,000 and Precision at 10. Normalized discounted cumulative gain (nDCG) is a measure for ranking quality that takes the ideal ranking of relevant documents into account. nDCG scores lie between 0.0 and 1.0. Higher scores are better. Precision at 10 describes the share of relevant documents in the 10 highest ranked retrieved documents.

The achieved scores of all three approaches are low, with an $nDCG@30 < 0.06$, an $nDCG@1,000 < 0.25$ and a $precision@10 < 0.07$. Even the baseline without any re-ranking achieves higher scores for all metrics. However, when comparing the three approaches, a clear distinction can be made. For nDCG@30 and Precision@10 and all three evaluated criteria, the approach utilizing only audio data achieves higher scores than the other two. Regarding "subjective" and "discussion", the combined approach ranks in second place. For the criterion "entertaining", the approach utilizing only text data and the combined approach achieve the same Precision@10 and almost equal nDCG@30 scores, with a slight edge for the approach utilizing only text.

		abstractive	extractive
Rouge1	Precision	0.216	0.14
	Recall	0.189	0.331
	F1-Score	0.17	0.164
RougeL	Precision	0.127	0.073
	Recall	0.12	0.196
	F1-Score	0.103	0.088

Table 4.3: Average Precision, Recall and F1-Score for Rouge1 and RougeL for the two proposed summarization approaches. Calculated in reference to the episode descriptions set by the creator.

4.3 Summarization

We evaluate the two proposed approaches for the generation of short text summaries of podcast episodes. For this, we firstly execute an automatic evaluation using Rouge scores. Secondly, we present a selection of generated summaries as examples. Lastly, we present the results of the manual assessments of the TREC 2021 podcast track.

Generated summaries from both approaches are compared to the episode descriptions set by the creators as a reference. Table 4.3 shows the results. We calculate average Precision, Recall and F1-Score for the metrics Rouge1 and RougeL. Rouge scores are utilized for the evaluation of automatic summarization systems. Rouge1 describes the overlap of unigrams between generated summaries and references. RougeL is based on the longest common subsequences between generated summaries and references.

The abstractive approach achieves higher Precision and F1-Scores than the extractive approach for both Rouge scores. However, the extractive approach achieves higher Recall for both.

We present two episodes and the corresponding generated summaries as examples. We also show the description of the episode as a reference.

Episode 1

Firstly, we summarize the episode "5 things to do during the COVID-19 Lockdown."[4] of the "integrate Podcast". This episode produces a bad and nonsensical abstractive summary, but an adequate extractive summary that contains

[4] `https://open.spotify.com/episode/2wED3vhTh7TgsQzGygvvkK`

some information about the contents of the episodes. However, many of the sentences are not comprehensible. The abstractive summary reads as follows:

> "There's a speculation that the lockdown was lifted, it's worth knowing that, we'll still have to mend in quarantine and there are very high, John says that we have to be at home. Dives deep into the mentality and computer side of the greatest athlete of time and G20 3/4 ton dual series which will give you Goosebumps 4 days."

The extrative summary reads as follows:

> "Anger is a free podcasting app that allows you to clean thousands of podcast from an assortment of genres. You, it's as powerful as any of the pinion broadcasting of sites out there. FM after this episode. Welcome to another episode of integrate to talk about from the title of this podcast. Anyhow. It can be you writing about the favorite game movie song, except the list is virtually endless number for Netflix. Has one of these masterpieces is the Last Dance, which is a biography on the life of Michael Jai. And last but not the least working out, this unfortunately has become a key to 45 minutes of exercise will elevate your mood and physical and mental state to a whole new level of working out doesn't mean lifting weights body weight, exercises are even meditation? Any other passion to? But if you're listening to this."

Additionally, the episode description reads as follows:

> "This episode suggests 5 productive and mentally engaging things to do during this difficult time to keep oneself physically and mentally healthy. — This episode is sponsored by Anchor: The easiest way to make a podcast. https://anchor.fm/app"

Episode 2

Secondly, we summarize the episode "Episode 4 part 2 Ron Tarquinio Story"[5] of the podcast "Near Falls With DHall". This episode produces a good abstractive summary that contains relevant information about the contents of the episode, but a bad extractive summary containing only incoherent sentences. The abstractive summary reads as follows:

> "Ron tarquinio wrestled at West Allegheny High School in Pittsburgh, Pennsylvania. He wrestled in high school and college wrestling. His senior Freight leaving off with going, through his senior freestyle bracket, and ultimately winning, and taking home, the stop sign."

[5] `https://open.spotify.com/episode/6YTHeAu9T5hOViLPMu8Rjo`

The extrative summary reads as follows:

> "You know what? I just remembered just being like excited, right? Yager so, you know how to beat Frankie? They got a whole roof and died with the sound of headgear makes with the crowd sound, like, when it's like, people are screaming like crazy, and I'm just laying, like, back back. I would have been in like a really good spot, right? Breathing was just like, had a long day, you going to wrestling, right? What was it like stepping back in the room knowing you know, You're going to be the the guy, you know? So you know, the kind of the kids are coming in and it's not like you're coming in with the cupboard bare, you know what I mean? You know what I listen? And I started doing with the same with Aaron, Give me your favorite near fall and or pain in your career because Mighty Mite, you know, my name is near Falls with the hall, it's the only the appropriate way to end, you know, in the episode in this way."

Additionally, the episode description reads as follows:

> "In the second part of the Ron Tarquinio story we get into Ron's college years at Pitt. We get into his ups like beating the the number 1 kid in the county. Also his downs like his grandfather and mentor passing away right after he qualified for nationals his senior year. We also touch on how he got into coaching and why he ultimately left West Allegheny."

4.3.1 TREC Manual Assessment

Both approaches were assessed in two ways in the manual assessment. As a first assessment, each generated summary is graded on a four point scale. The grades are "Excellent" (E), "Good" (G), "Fair" (F) and "Bad" (B). Additionally, we calculate an aggregated score by assigning $E = 4$, $G = 2$, $F = 1$, $B = 0$ and calculating the average value. Results are shown in table 4.4. The extractive approach achieves slightly better results than the abstractive approach. Both generate 6 summaries assessed as "good", but 38 summaries of the extractive approach are graded as "Fair", five more than the 33 of the abstractive approach. The rest of the summaries are graded as "bad". Subsequently, the extractive approach achieves a slightly higher aggregated score of 0.2604, in contrast to the 0.2332 of the abstractive approach. However, both approaches achieve significantly lower scores than a simple baseline, which utilizes the first minute of the episode transcript as a summary.

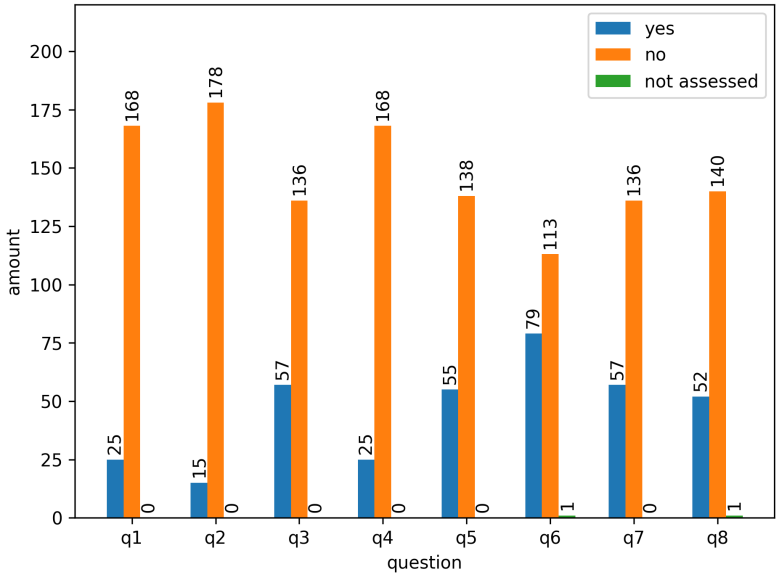
run	aggregate EGFB score	#E	#G	#F	#B	not rated
abstractive	0.2332	0	6	33	154	0
extractive	0.2604	0	6	38	148	1
baseline (one-minute)	0.8083	7	26	76	84	0

Table 4.4: Overview of manual assessment results for 193 summarized episodes. The aggregate EGFB score is computed by calculating the average while assigning E=4, G=2, F=1, B=0. A baseline is shown for comparison. The baseline utilizes the transcript of the first minute of the episode as the summary.

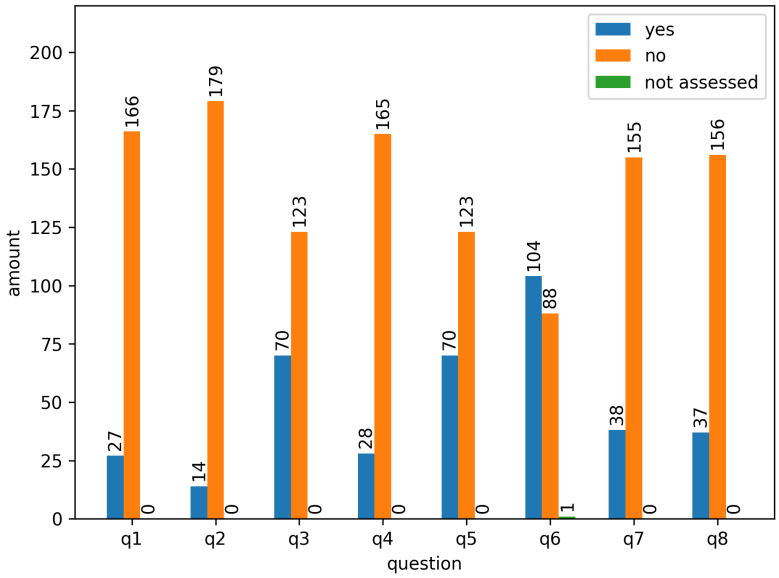
As a second assessment, 8 questions about each summary are answered by TREC assessors. The answers of the questions are presented in figure 4.2. The questions are as follows:

1. Does the summary include names of the main people (hosts, guests, characters) involved or mentioned in the podcast?
2. Does the summary give any additional information about the people mentioned (such as their job titles, biographies, personal background, etc)?
3. Does the summary include the main topic(s) of the podcast?
4. Does the summary tell you anything about the format of the podcast; e.g. whether it is an interview, whether it is a chat between friends, a monologue, etc?
5. Does the summary give you more context on the title of the podcast?
6. Does the summary contain redundant information?
7. Is the summary written in good English?
8. Are the start and end of the summary good sentence and paragraph start and end points?

Especially noticeable are the results of question 6 for the extractive summary, as more than half of the summaries contain redundant information. Furthermore, both approaches miss information about the involved people or the format of the podcast in most of the summaries, as the results of questions 1, 2 and 4 show. However, question 3 shows that 29.5% of abstractive and 36.3% of extractive summaries at least contain information about the main topics of the episodes.

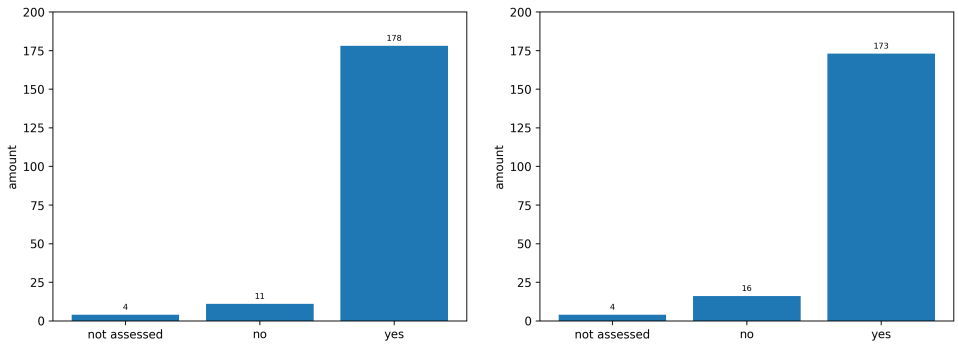


(a) Abstractive approach



(b) Extractive approach

Figure 4.2: Results of eight "yes"/"no" questions (q1-q8) asked for each of the 193 episode summaries about their content. The questions are presented in section 4.3.1. Some questions were not answered by the assessors for some episodes. They are marked as "not assessed".



(a) Anchor Points **(b)** TextRank

Figure 4.3: Manual evaluation of the 193 one minute audio clips. Assessors where asked the following question after listening: "Does the clip give a sense of what the podcast sounds like, (as far as you can tell from listening to it)?". Some audio clips were not assessed.

4.4 Audio Clips

We evaluate the two approaches for the creation of audio clips of a duration of up to one minute. For this, we utilize the manual assessments from the TREC 2021 podcast track.

4.4.1 TREC Manual Assessment

A total of 193 audio clips for each of the two approaches were submitted for manual assessment. The assessors listen to a podcast episode and the corresponding audio clip. Afterwards, they answer the following question with "Yes" or "No":

> "Does the clip give a sense of what the podcast sounds like, (as far as you can tell from listening to it)?"

For each approach, 4 episodes were not assessed. However, these 4 episodes do not completely overlap.

Evaluation results for the approach using entertaining anchor points, based on the proposed abstractive summarization approach, are presented in figure 4.3a. The approach achieves good results. 94% of assessed audio clips received a positive assessment. Based on this, the approach generally can be seen as suitable for the task of giving listeners a general first impression of a podcast.

Evaluation results for the approach using the TextRank algorithm, based on the proposed extractive summarization approach, are presented in figure 4.3b.

The approach also achieves good results. 92% of assessed audio clips received a positive assessment, which is only slightly worse than the other approach. However, the approach still seems to be suitable for the task of giving listeners a general first impression of a podcast as well.

Chapter 5

Discussion

In this chapter, we discuss the results of this thesis and the evaluated approaches. We also present the results of our submitted approaches in the TREC 2021 podcast track in comparison to approaches by other participants.

As shown in the confusion matrices in section 4.1, the binary re-ranking criteria classifications generally perform adequately on our own data during the cross validation for all three criteria. Especially the classification of the criterion "discussion" performs well. This might be explained by the relative unambiguity of the criterion compared to "entertaining" and "subjective", which may result in better training data as assessors are less likely to reach inconsistent results when annotating segments. However, when applying the classifications to the three re-ranking tasks, the results are quite bad with very low nDCG and Precision scores. This could be due to several reasons. One possible explanation is that the conversion of binary classifications to a float value between 0.0 and 1.0 is not working as intended. Another explanation might be inadequate generalization of the classifying models. Moreover, the long duration of the classified segments of two minutes might be the reason, as the dimensionality of the inputs may be too high. A segmentation into multiple shorter segments and some sort of aggregation on their classifications may be beneficial. Additionally, badly annotated training data could be the reason. Annotators might have had a wrong understanding of the criteria, which conflicted with the assessments of the TREC annotators. Further investigations on causes for the bad classification results are needed.

Furthermore, the general results of the evaluation of both summarization approaches are disappointing as well. Both approaches perform worse than even a simple baseline of the first minute of the episode transcript. It should be noted, however, that the generated short audio clips of both approaches achieve good scores. This might be due to looser requirements for the evaluation. However, another possible explanation is that the audio clips do not rely

on automatically created transcriptions, which do not produce perfect transcripts. Many of the generated text summaries by both proposed approaches are assessed as not being expressed in good English (see figure 4.2). Especially concerning the extractive approach, this is likely because of bad transcription quality, as the system utilizes text directly from the transcript. It should also be noted that our summarization approaches intend to create summaries that arouse interest of potential listeners. However, this aspect is not included in the manual assessments at all.

Generally, the evaluation of the summarization approaches is difficult. Both approaches are based largely on the usage of entertaining sentences. Due to the bad classification performance, the approaches can not operate as intended, as the underlying classifications are not reliable. We therefore can not draw direct conclusions on the performance of the approaches themself. Further investigations on the performance of the summarization models not dependent on the performance of the classification are needed. For this, a manual classification of entertaining sentences may be helpful as a basis for an evaluation.

This thesis intends to study the effects of the usage of audio data for the classification of podcast episodes, in contrast to only utilizing transcripts. For this, we compare the results of the evaluation of the proposed systems utilizing only text data, only audio data or a combination of both. The evaluation of the predictions from the cross validation in section 4.1 demonstrates that the approach utilizing a combination of text and audio data generally performs the best. However, we also compare the performance of the re-ranking approaches utilizing audio data, text data and a combination of both. As described in section 4.2.1, the re-ranking approach utilizing audio data achieves higher scores for all three criteria than the approach utilizing only text data in the TREC evaluation.

According to our evaluation, the addition of audio data benefits the classification of the three investigated criteria in comparison to only utilizing text data. Considering the types of utilized criteria (entertaining, subjective and discussion), our findings are plausible. Especially the criteria "entertaining" and "discussion" are often characterized by features that are largely not contained in the text data. For example, entertaining segments might be characterized by laughter or the tone of the voice of the speakers. Both of which are not apparent in the text data. Additionally, for the classification of the criterion "discussion", the benefit of utilizing audio data is obvious, as the presence of multiple different voices can be recognized rather easily in audio, in contrast to a text.

5.1 TREC 2021 Podcast Track

We submitted all proposed approaches in the 2021 iteration of the TREC podcast track. This includes four approaches for segment retrieval (focused on re-ranking), two summarization approaches and two approaches for the creation of short audio clips. At the time of this thesis, only a preliminary overview paper by Karlgren et al. [2021] describing the results of the the track was released. A final version is expected in 2022. Hereinafter, we briefly present the scores of our approaches in comparison to the approaches of other participants.

The tables 5.1 display the results of all participants in the three re-ranking components of the segment retrieval task. When excluding our baseline, our approaches are ranked last for all three criteria. However, all submitted experiments of all participants result in generally low scores, especially for the criteria "entertaining" and "discussion".

Table 5.2 shows the results of the manual assessment of the summarization systems. Our two approaches achieve lower scores than all but one of the other approaches. However, the scores of all submitted systems are generally low. The best approach receives a score of only 1.06, which is out of a maximum of 4.0. No approach performs significantly better than the baseline. Generally it can be seen that the six best performing approaches all generate abstractive summaries.

Table 5.3 shows the results of the manual assessment of the generated audio clips. Our two approaches achieve comparable results to most other runs, but are still slightly worse than the baseline. In general, most systems perform well in this task with one system even achieving the maximum possible score of 1.0.

	nDCG	nDCG at 30	precision at 10
tp_mt5	**0.30**	**0.20**	**0.18**
TUW_hybrid_cat	**0.31**	**0.16**	**0.15**
osc_vec_tok	0.19	**0.16**	**0.15**
tp_mt5_f1	0.27	**0.16**	**0.14**
osc_tok_vec	0.19	**0.16**	**0.14**
TUW_tasb_cat	0.27	0.15	**0.14**
f_b25_tct	**0.30**	0.15	**0.13**
TUW_hybrid_ws	**0.29**	0.15	**0.13**
f_coil_tct	0.27	0.15	**0.13**
tp_mt5_f2	0.26	**0.16**	**0.12**
ms_mt5	0.27	0.14	**0.12**
s_tct	0.23	0.14	**0.12**
TUW_tasb192_ann	0.23	0.14	**0.12**
osc_vector	0.15	0.12	**0.12**
f_b25_coil	0.27	0.13	0.11
osc_token	0.19	0.15	0.10
Webis_pc_bs	0.23	0.12	0.10
s_tasb	0.22	0.12	0.10
UCL_audio_2	0.15	0.11	0.08
UCL_audio_1	0.15	0.10	0.07
Webis_pc_cola	0.17	0.05	0.05
Webis_pc_rob	0.16	0.04	0.03
Webis_pc_co_rob	0.16	0.03	0.03
Baseline BM25-D	0.24	0.11	0.09
Baseline QL-D	0.25	0.11	0.09
Baseline BM25-Q	0.26	0.14	0.10
Baseline QL-Q	0.27	0.14	0.10

(a) Entertaining

	nDCG	nDCG at 30	precision at 10
tp_mt5	**0.47**	**0.31**	**0.35**
tp_mt5_f1	**0.46**	**0.31**	**0.33**
f_b25_tct	**0.49**	**0.29**	**0.32**
osc_vec_tok	0.32	**0.28**	**0.32**
osc_tok_vec	0.31	**0.27**	**0.32**
TUW_tasb192_ann	0.39	**0.25**	**0.32**
tp_mt5_f2	**0.46**	**0.30**	**0.31**
f_coil_tct	**0.46**	**0.28**	**0.31**
s_tct	0.41	**0.27**	**0.29**
TUW_hybrid_ws	**0.48**	**0.27**	**0.29**
TUW_tasb_cat	**0.46**	**0.27**	**0.28**
f_b25_coil	**0.45**	**0.24**	**0.28**
TUW_hybrid_cat	**0.49**	**0.26**	**0.27**
osc_vector	0.28	0.23	**0.27**
s_tasb	0.37	0.22	**0.27**
ms_mt5	0.41	0.22	0.23
osc_token	0.27	0.21	0.23
UCL_audio_1	0.25	0.19	0.20
UCL_audio_2	0.24	0.19	0.20
Webis_pc_bs	0.34	0.17	0.20
Webis_pc_cola	0.24	0.06	0.06
Webis_pc_co_rob	0.23	0.04	0.06
Webis_pc_rob	0.22	0.04	0.04
Baseline BM25-D	0.40	0.21	0.24
Baseline QL-D	0.41	0.21	0.24
Baseline BM25-Q	0.40	0.22	0.22
Baseline QL-Q	0.41	0.22	0.21

(b) Subjective

	nDCG	nDCG at 30	precision at 10
f_b25_tct	**0.43**	**0.24**	**0.22**
f_coil_tct	**0.41**	**0.24**	**0.22**
osc_tok_vec	0.27	**0.22**	**0.22**
tp_mt5	**0.39**	**0.22**	**0.21**
tp_mt5_f1	0.38	**0.22**	**0.21**
tp_mt5_f2	0.38	0.21	**0.21**
TUW_tasb192_ann	0.34	0.20	**0.21**
TUW_hybrid_ws	**0.44**	**0.24**	0.20
osc_vec_tok	0.28	**0.23**	0.20
f_b25_coil	**0.40**	0.21	0.20
s_tct	0.35	**0.22**	0.19
TUW_tasb_cat	**0.39**	0.21	0.19
s_tasb	0.34	0.20	0.19
osc_vector	0.24	0.19	0.19
TUW_hybrid_cat	**0.42**	0.21	0.18
ms_mt5	0.36	0.18	0.17
osc_token	0.25	0.19	0.16
Webis_pc_bs	0.32	0.16	0.16
UCL_audio_2	0.25	0.19	0.15
UCL_audio_1	0.24	0.19	0.15
Webis_pc_cola	0.23	0.06	0.06
Webis_pc_co_rob	0.22	0.05	0.06
Webis_pc_rob	0.21	0.04	0.04
Baseline BM25-D	0.36	0.19	0.18
Baseline QL-D	0.37	0.18	0.19
Baseline BM25-Q	0.37	0.20	0.16
Baseline QL-Q	0.37	0.20	0.15

(c) Discussion

Table 5.1: Overview of ranked results for the submitted experiments of the TREC 2021 podcast track for the re-ranking of retrieved segments [Karlgren et al., 2021]. Presented metrics are nDCG, nDCG@30 and Precision@10. Our four submitted experiments are "Webis_pc_bs" (baseline), "Webis_pc_cola" (audio), "Webis_pc_rob" (text) and "Webis_pc_co_rob" (combined).

experiment	type	summarization quality
PoliTO_50_64-128	A	1.06
Unicamp1	A	1.04
PoliTO_25_32-128	A	1.03
Unicamp2	A	1.01
PoliTO_100_32-128	A	0.98
PoliTO_50_32-128	A	0.91
Baseline onemin	E	0.81
Hotspot1	E	0.43
theTuringTest1	E	0.34
Webis_pc_extr	**E**	**0.26**
Webis_pc_abstr	**A**	**0.23**
theTuringTest2	A	0.18

Table 5.2: Overview of ranked results for the submitted experiments of the TREC 2021 podcast track for summarization [Karlgren et al., 2021]. Our two submitted approaches are typed in bold. The abstractive approach is called "Webis_pc_abstr". The extractive approach is called "Webis_pc_extr". The type specifies whether an approach is abstractive or extractive. The quality ratings (EGFB scores) range from 0.0 to 4.0.

experiment	audio clip assessment
Unicamp1	1.00
PoliTO_100_32-128	0.99
PoliTO_50_32-128	0.99
PoliTO_50_64-128	0.98
PoliTO_25_32-128	0.98
Baseline onemin	0.96
Hotspot1	0.95
Webis_pc_abstr	**0.94**
Webis_pc_extr	**0.92**
Unicamp2	0.50
theTuringTest2	0.21
theTuringTest1	0.20

Table 5.3: Overview of ranked results for the submitted experiments of the TREC 2021 podcast track for the created audio clips [Karlgren et al., 2021]. Our two submitted approaches are typed in bold. The approach utilizing entertaining anchor points is called "Webis_pc_abstr". The approach utilizing TextRank is called "Webis_pc_extr". The scores range from 0.0 to 1.0 and are the results of one question ("Does the clip give a sense of what the podcast sounds like, (as far as you can tell from listening to it)?") answered by the assessors for each audio clip.

Chapter 6

Conclusion

We demonstrated a slight benefit of utilizing audio data for the classification of podcast segments for the criteria "entertaining", "subjective" and "discussion" in comparison to only utilizing text data. However, the general performance of the classification in the context of re-ranking podcast segments and as the basis for the automatic creation of podcast summaries still has a lot of potential for improvement. We also presented approaches for the automatic creation of short text summaries and short audio clips for podcast episodes.

As discussed in section 5, as a next step it is necessary to further investigate the reasons for the bad performance of the classifications. For this, a detailed analysis of classified segments can be conducted to verify which kind of segments the approaches are able to classify well and which kind of segments lead to issues. This information can be utilized to enhance the systems, for example by adjusting and enhancing the utilized training data. Furthermore, the annotated training set can be revised by additional annotators to ensure a consensus on selected labels for the three criteria for each annotated segment. Afterwards, all models need to be trained again on the revised training data and need to be evaluated once again to investigate whether this leads to an improvement. Another approach would be to utilize the released criteria annotations of the TREC 2021 podcast track and cross-validate the classifiers on these. Thus we are able to verify whether a different understanding of the criteria by the annotators had an impact on the resulting outcome. The impact of the duration of classified segments should also be studied. Different lengths of segments and the resulting performance of the classifiers should be compared. If short segments are classified more successfully, long segments can be split into multiple shorter segments to classify and the resulting classifications are aggregated, e.g. as an average value. Additionally, a further evaluation of the summarization approaches independent of the performance of the criteria classification can be conducted by utilizing manually labeled sentences instead

of automatically classified sentences.

Although the TREC podcast track has been paused following the 2021 iteration, the domains of podcast retrieval and podcast summarization still offer many possibilities for future research.

Appendix A

100 Queries Used For Manually Annotated Segments

1. coronavirus spread
2. greta thunberg cross atlantic
3. black hole image
4. story about riding a bird
5. daniel ek interview
6. michelle obama becoming
7. anna delvey
8. facebook stock prediction
9. trump call ukrainian president
10. boeing 737 crash causes
11. how to cook turkey
12. imran khan career
13. drug addiction recovery
14. near death experiences
15. podcast about podcasts
16. causes and prevention of wildfires

17. time between meetings
18. women in stem
19. ai in healthcare
20. cost of childcare
21. juneteenth
22. chernobyl hbo
23. notre dame fire
24. france yellow vest protests
25. black lives matter
26. bob woodward
27. civil rights protest stories
28. yo-yo dieting
29. racism in canada
30. motherhood
31. horoscope reading cancer
32. giants game december 22
33. hvac industry environmentalism
34. halloween stories and chat
35. living debt free
36. cryptocurrency risks
37. slow travel
38. workplace diversity
39. social media marketing
40. bees dying
41. gmo food labeling

42. fyre festival
43. hong kong protests
44. thanksgiving comedy special
45. drafting tight ends
46. missouri quilt mom
47. sci-fi author interview mars
48. spike lee movie score
49. thrift store smell
50. coast guard coxswain
51. queer eye veteran
52. fauci interview
53. recommended books for entrepreneurs
54. bias in college admissions
55. malcolm x biography
56. gaslighting
57. nefertiti
58. sam bush interview
59. adopting a dog
60. $23 million lottery winner forgot
61. first face transplant
62. amc breaking bad
63. does gun control reduce gun violence
64. france world cup 1998
65. difference between mitosis and meiosis
66. child development cultural differences

67. heart attack statistics
68. how to get fit safely
69. advantages of junk food
70. furniture for small spaces
71. confederations cup 2013
72. advice for technology start-up
73. 42 years in a coma
74. $425 million jackpot
75. cuban offers trump $1 million
76. depression symptoms
77. earn money at home
78. earthquake hits japan
79. einstein relativity theory explained
80. gift ideas for college students
81. girl slept for 64 days
82. god does not exist
83. harry potter and the goblet of fire book
84. help retirement plan
85. how to get a pay raise
86. how to quit smoking
87. indian wedding culture rituals
88. laser treatment beauty safety
89. latest lakers rumors
90. march madness semi finals
91. obama family tree

92. pink floyd the wall review
93. running a half marathon
94. world war z movie review
95. lionel messi vs cristiano ronaldo
96. bojack horseman new season
97. how to earn money with a podcast
98. angela merkel election
99. nba rule change
100. is bitcoin the future

Bibliography

Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer. *CoRR*, abs/2004.05150, 2020. URL `https://arxiv.org/abs/2004.05150`.

Michael Bendersky, Donald Metzler, and W. Bruce Croft. Learning concept importance using a weighted dependence model. In Brian D. Davison, Torsten Suel, Nick Craswell, and Bing Liu, editors, *Proceedings of the Third International Conference on Web Search and Web Data Mining, WSDM 2010, New York, NY, USA, February 4-6, 2010*, pages 31–40. ACM, 2010. doi: 10.1145/1718487.1718492. URL `https://doi.org/10.1145/1718487.1718492`.

Oberon Berlage, Klaus-Michael Lux, and David Graus. Improving automated segmentation of radio shows with audio embeddings. *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2020. doi: 10.1109/icassp40776.2020.9054315. URL `http://dx.doi.org/10.1109/ICASSP40776.2020.9054315`.

Jana Besser, Katja Hofmann, and Martha Larson. An exploratory study of user goals and strategies in podcast search. In *In Proceedings of the FGIR Workshop Information Retrieval (WIR2008)*, pages 3–10, 2008.

Christoph Brachmann, Hashim Chunpir, Silke Gennies, Benjamin Haller, Philipp Kehl, Astrid Mochtarram, Daniel Möhlmann, Christian Schrumpf, Christopher Schultz, Björn Stolper, Benjamin Walther-Franks, Arne Jacobs, Thorsten Hermes, and Otthein Herzog. *Automatic Movie Trailer Generation Based on Semantic Video Patterns*, pages 145–158. 12 2009. ISBN 9783837610239. doi: 10.14361/9783839410233-011.

Sylvia Chan-Olmsted and Rang Wang. Understanding podcast users: Consumption motives and behaviors. *New Media and Society*, page 146144482096377, 10 2020. doi: 10.1177/1461444820963776.

Ann Clifton, Sravana Reddy, Yongze Yu, Aasish Pappu, Rezvaneh Rezapour, Hamed Bonab, Maria Eskevich, Gareth Jones, Jussi Karlgren, Ben

Carterette, and Rosie Jones. 100,000 podcasts: A spoken english document corpus. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5903–5917, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics. URL https://www.aclweb.org/anthology/2020.coling-main.519.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 8440–8451. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.acl-main.747. URL https://doi.org/10.18653/v1/2020.acl-main.747.

Dennis. How to write better podcast show notes (with 3 templates). https://castos.com/podcast-show-notes/, 2021. Accessed: 11.01.22.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018. URL http://arxiv.org/abs/1810.04805.

Edison Research. The infinite dial 2020. https://www.edisonresearch.com/the-infinite-dial-2020/, 2020. Accessed: 04.01.22.

Günes Erkan and Dragomir R. Radev. Lexrank: Graph-based lexical centrality as salience in text summarization. *J. Artif. Intell. Res.*, 22:457–479, 2004. doi: 10.1613/jair.1523. URL https://doi.org/10.1613/jair.1523.

Petra Galuscáková, Suraj Nair, and Douglas W. Oard. Combine and re-rank: The university of maryland at the TREC 2020 podcasts track. In Ellen M. Voorhees and Angela Ellis, editors, *Proceedings of the Twenty-Ninth Text REtrieval Conference, TREC 2020, Virtual Event [Gaithersburg, Maryland, USA], November 16-20, 2020*, volume 1266 of *NIST Special Publication*. National Institute of Standards and Technology (NIST), 2020. URL https://trec.nist.gov/pubs/trec29/papers/UMD_IR.P.pdf.

Go Irie, Takashi Satou, Akira Kojima, Toshihiko Yamasaki, and Kiyoharu Aizawa. Automatic trailer generation. In *Proceedings of the 18th ACM International Conference on Multimedia*, MM '10, pages 839–842, New York, NY, USA, 2010. Association for Computing Machinery. ISBN 9781605589336. doi: 10.1145/1873951.1874092. URL https://doi.org/10.1145/1873951.1874092.

Rosie Jones, Ben Carterette, Ann Clifton, Jussi Karlgren, Aasish Pappu, Sravana Reddy, Yongze Yu, Maria Eskevich, and Gareth J. F. Jones. TREC 2020 podcasts track overview. In Ellen M. Voorhees and Angela Ellis, editors, *Proceedings of the Twenty-Ninth Text REtrieval Conference, TREC 2020, Virtual Event [Gaithersburg, Maryland, USA], November 16-20, 2020*, volume 1266 of *NIST Special Publication*. National Institute of Standards and Technology (NIST), 2020. URL `https://trec.nist.gov/pubs/trec29/papers/OVERVIEW.P.pdf`.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. Spanbert: Improving pre-training by representing and predicting spans. *CoRR*, abs/1907.10529, 2019. URL `http://arxiv.org/abs/1907.10529`.

Hannes Karlbom and Ann Clifton. Abstract podcast summarization using BART with longformer attention. In Ellen M. Voorhees and Angela Ellis, editors, *Proceedings of the Twenty-Ninth Text REtrieval Conference, TREC 2020, Virtual Event [Gaithersburg, Maryland, USA], November 16-20, 2020*, volume 1266 of *NIST Special Publication*. National Institute of Standards and Technology (NIST), 2020. URL `https://trec.nist.gov/pubs/trec29/papers/hk_uu_podcast.P.pdf`.

Jussi Karlgren, Rosie Jones, Ben Carterette, Ann Clifton, Maria Eskevich, Gareth J. F. Jones, Sravana Reddy, and Edgar Tanaka. TREC 2021 podcasts track overview. 2021. (Notebook version: final version to appear in the TREC proceedings in early 2022).

Sumanta Kashyapi and Laura Dietz. TREMA-UNH at TREC 2020. In Ellen M. Voorhees and Angela Ellis, editors, *Proceedings of the Twenty-Ninth Text REtrieval Conference, TREC 2020, Virtual Event [Gaithersburg, Maryland, USA], November 16-20, 2020*, volume 1266 of *NIST Special Publication*. National Institute of Standards and Technology (NIST), 2020. URL `https://trec.nist.gov/pubs/trec29/papers/TREMA_UNH.P.pdf`.

Vaibhav Kasturia, Marcel Gohsen, and Matthias Hagen. Query interpretations from entity-linked segmentations, 2022.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL `http://arxiv.org/abs/1412.6980`.

Oleksandra Klymenko, Daniel Braun, and Florian Matthes. Automatic text summarization: A state-of-the-art review. In Joaquim Filipe, Michal Smialek, Alexander Brodsky, and Slimane Hammoudi, editors, *Proceedings of the 22nd International Conference on Enterprise Information Systems, ICEIS 2020, Prague, Czech Republic, May 5-7, 2020, Volume 1*, pages 648–655. SCITEPRESS, 2020. doi: 10.5220/0009723306480655. URL https://doi.org/10.5220/0009723306480655.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7871–7880. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.acl-main.703. URL https://doi.org/10.18653/v1/2020.acl-main.703.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019. URL http://arxiv.org/abs/1907.11692.

Potsawee Manakul and Mark J. F. Gales. Cued_speech at TREC 2020 podcast summarisation track. In Ellen M. Voorhees and Angela Ellis, editors, *Proceedings of the Twenty-Ninth Text REtrieval Conference, TREC 2020, Virtual Event [Gaithersburg, Maryland, USA], November 16-20, 2020*, volume 1266 of *NIST Special Publication*. National Institute of Standards and Technology (NIST), 2020. URL https://trec.nist.gov/pubs/trec29/papers/cued_speech.P.pdf.

Potsawee Manakul, Mark J. F. Gales, and Linlin Wang. Abstractive spoken document summarization using hierarchical model with multi-stage attention diversity optimization. In Helen Meng, Bo Xu, and Thomas Fang Zheng, editors, *Interspeech 2020, 21st Annual Conference of the International Speech Communication Association, Virtual Event, Shanghai, China, 25-29 October 2020*, pages 4248–4252. ISCA, 2020. doi: 10.21437/Interspeech.2020-1683. URL https://doi.org/10.21437/Interspeech.2020-1683.

Mark. What is dynamic ad insertion in podcasting? https://www.captivate.fm/blog/what-is-dynamic-ad-insertion/, 2021. Accessed: 09.02.22.

Matthew McLean. Podcast sponsorship – everything you need to know. https://www.thepodcasthost.com/monetisation/how-to-do-podcast-sponsorship/, 2021. Accessed: 09.01.22.

Rada Mihalcea and Paul Tarau. TextRank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL https://aclanthology.org/W04-3252.

Derek Miller. Leveraging BERT for extractive text summarization on lectures. *CoRR*, abs/1906.04165, 2019. URL http://arxiv.org/abs/1906.04165.

Dan Misener. Podcast episodes got shorter in 2019. https://blog.pacific-content.com/podcast-episodes-got-shorter-in-2019-69e1f3b6c82f, 2019. Accessed: 11.09.21.

Yasufumi Moriya and Gareth J. F. Jones. DCU-ADAPT at the TREC 2020 podcasts track. In Ellen M. Voorhees and Angela Ellis, editors, *Proceedings of the Twenty-Ninth Text REtrieval Conference, TREC 2020, Virtual Event [Gaithersburg, Maryland, USA], November 16-20, 2020*, volume 1266 of *NIST Special Publication*. National Institute of Standards and Technology (NIST), 2020. URL https://trec.nist.gov/pubs/trec29/papers/DCU-ADAPT.P.pdf.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. *CoRR*, abs/1808.08745, 2018. URL http://arxiv.org/abs/1808.08745.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. MS MARCO: A human generated machine reading comprehension dataset. In Tarek Richard Besold, Antoine Bordes, Artur S. d'Avila Garcez, and Greg Wayne, editors, *Proceedings of the Workshop on Cognitive Computation: Integrating neural and symbolic approaches 2016 co-located with the 30th Annual Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, December 9, 2016*, volume 1773 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2016. URL http://ceur-ws.org/Vol-1773/CoCoNIPS_2016_paper9.pdf.

Rodrigo Nogueira and Kyunghyun Cho. Passage re-ranking with BERT. *CoRR*, abs/1901.04085, 2019. URL http://arxiv.org/abs/1901.04085.

Paul Owoicho and Jeff Dalton. Glasgow representation and information learning lab (GRILL) at TREC 2020 podcasts track. In Ellen M. Voorhees and Angela Ellis, editors, *Proceedings of the Twenty-Ninth Text REtrieval Conference, TREC 2020, Virtual Event [Gaithersburg, Maryland, USA], November 16-20, 2020*, volume 1266 of *NIST Special Publication*. National Institute of Standards and Technology (NIST), 2020. URL `https://trec.nist.gov/pubs/trec29/papers/uog_msc.P.pdf`.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12: 2825–2830, 2011.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *CoRR*, abs/1910.10683, 2019. URL `http://arxiv.org/abs/1910.10683`.

Sravana Reddy, Mariya Lazarova, Yongze Yu, and Rosie Jones. Modeling language usage and listener engagement in podcasts. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 632–643. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.acl-long.52. URL `https://doi.org/10.18653/v1/2021.acl-long.52`.

Rezvaneh Rezapour, Sravana Reddy, Ann Clifton, and Rosie Jones. Spotify at TREC 2020: Genre-aware abstractive podcast summarization. In Ellen M. Voorhees and Angela Ellis, editors, *Proceedings of the Twenty-Ninth Text REtrieval Conference, TREC 2020, Virtual Event [Gaithersburg, Maryland, USA], November 16-20, 2020*, volume 1266 of *NIST Special Publication*. National Institute of Standards and Technology (NIST), 2020. URL `https://trec.nist.gov/pubs/trec29/papers/Spotify.P2.pdf`.

Stephen E. Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. Okapi at TREC-3. In Donna K. Harman, editor, *Proceedings of The Third Text REtrieval Conference, TREC 1994, Gaithersburg, Maryland, USA, November 2-4, 1994*, volume 500-225 of *NIST Special Publication*, pages 109–126. National Institute of Standards

and Technology (NIST), 1994. URL `http://trec.nist.gov/pubs/trec3/papers/city.ps.gz`.

Aaqib Saeed, David Grangier, and Neil Zeghidour. Contrastive learning of general-purpose audio representations. *CoRR*, abs/2010.10915, 2020. URL `https://arxiv.org/abs/2010.10915`.

Abigail See, Peter J. Liu, and Christopher D. Manning. Get to the point: Summarization with pointer-generator networks. In Regina Barzilay and Min-Yen Kan, editors, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1073–1083. Association for Computational Linguistics, 2017. doi: 10.18653/v1/P17-1099. URL `https://doi.org/10.18653/v1/P17-1099`.

Abheesht Sharma and Harshit Pandey. LRG at TREC 2020: Document ranking with xlnet-based models. In Ellen M. Voorhees and Angela Ellis, editors, *Proceedings of the Twenty-Ninth Text REtrieval Conference, TREC 2020, Virtual Event [Gaithersburg, Maryland, USA], November 16-20, 2020*, volume 1266 of *NIST Special Publication*. National Institute of Standards and Technology (NIST), 2020. URL `https://trec.nist.gov/pubs/trec29/papers/LRG_REtrievers.P.pdf`.

Matthew Sharpe. A review of metadata fields associated with podcast RSS feeds. *CoRR*, abs/2009.12298, 2020. URL `https://arxiv.org/abs/2009.12298`.

Sam Shleifer and Alexander M. Rush. Pre-trained summarization distillation. *CoRR*, abs/2010.13002, 2020. URL `https://arxiv.org/abs/2010.13002`.

Joel Shor, Aren Jansen, Ronnie Maor, Oran Lang, Omry Tuval, Félix de Chaumont Quitry, Marco Tagliasacchi, Ira Shavitt, Dotan Emanuel, and Yinnon Haviv. Towards learning a universal non-semantic representation of speech. In Helen Meng, Bo Xu, and Thomas Fang Zheng, editors, *Interspeech 2020, 21st Annual Conference of the International Speech Communication Association, Virtual Event, Shanghai, China, 25-29 October 2020*, pages 140–144. ISCA, 2020. doi: 10.21437/Interspeech.2020-1242. URL `https://doi.org/10.21437/Interspeech.2020-1242`.

Kaiqiang Song, Fei Liu, Chen Li, Xiaoyang Wang, and Dong Yu. Automatic summarization of open-domain podcast episodes. In Ellen M. Voorhees and Angela Ellis, editors, *Proceedings of the Twenty-Ninth Text*

REtrieval Conference, TREC 2020, Virtual Event [Gaithersburg, Maryland, USA], November 16-20, 2020, volume 1266 of *NIST Special Publication*. National Institute of Standards and Technology (NIST), 2020. URL `https://trec.nist.gov/pubs/trec29/papers/UCF_NLP.P.pdf`.

Josef Steinberger and Karel Jezek. Using latent semantic analysis in text summarization and summary evaluation. 2004.

Marco Tagliasacchi, Beat Gfeller, Felix de Chaumont Quitry, and Dominik Roblek. Self-supervised audio representation learning for mobile devices. *CoRR*, abs/1905.11796, 2019. URL `http://arxiv.org/abs/1905.11796`.

Marco Tagliasacchi, Beat Gfeller, Felix de Chaumont Quitry, and Dominik Roblek. Pre-training audio representations with self-supervision. *IEEE Signal Process. Lett.*, 27:600–604, 2020. doi: 10.1109/LSP.2020.2985586. URL `https://doi.org/10.1109/LSP.2020.2985586`.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

Till von Wenzlawowicz and Otthein Herzog. Semantic video abstracting: Automatic generation of movie trailers based on video patterns. In Ilias Maglogiannis, Vassilis P. Plagianakos, and Ioannis P. Vlahavas, editors, *Artificial Intelligence: Theories and Applications - 7th Hellenic Conference on AI, SETN 2012, Lamia, Greece, May 28-31, 2012. Proceedings*, volume 7297 of *Lecture Notes in Computer Science*, pages 345–352. Springer, 2012. doi: 10.1007/978-3-642-30448-4_44. URL `https://doi.org/10.1007/978-3-642-30448-4_44`.

Ross Winn. 2021 podcast stats & facts (new research from apr 2021). `https://www.podcastinsights.com/podcast-statistics/`, 2021. Accessed: 08.09.21.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System*

Demonstrations, pages 38–45, Online, October 2020. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.emnlp-demos.6.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. Xlnet: Generalized autoregressive pretraining for language understanding. *CoRR*, abs/1906.08237, 2019. URL http://arxiv.org/abs/1906.08237.

Yongze Yu, Jussi Karlgren, Ann Clifton, Md. Iftekhar Tanveer, Rosie Jones, and Hamed Bonab. Spotify at the TREC 2020 podcasts track: Segment retrieval. In Ellen M. Voorhees and Angela Ellis, editors, *Proceedings of the Twenty-Ninth Text REtrieval Conference, TREC 2020, Virtual Event [Gaithersburg, Maryland, USA], November 16-20, 2020*, volume 1266 of *NIST Special Publication*. National Institute of Standards and Technology (NIST), 2020. URL https://trec.nist.gov/pubs/trec29/papers/Spotify.P.pdf.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. PEGASUS: pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 11328–11339. PMLR, 2020. URL http://proceedings.mlr.press/v119/zhang20ae.html.

Chujie Zheng, Harry Jiannan Wang, Kunpeng Zhang, and Ling Fan. A baseline analysis for podcast abstractive summarization. *CoRR*, abs/2008.10648, 2020a. URL https://arxiv.org/abs/2008.10648.

Chujie Zheng, Harry Jiannan Wang, Kunpeng Zhang, and Ling Fan. A Two-Phase Approach for Abstractive Podcast Summarization. In Ellen M. Voorhees and Angela Ellis, editors, *Proceedings of the Twenty-Ninth Text REtrieval Conference, TREC 2020, Virtual Event [Gaithersburg, Maryland, USA], November 16-20, 2020*, volume 1266 of *NIST Special Publication*. National Institute of Standards and Technology (NIST), 2020b. URL https://trec.nist.gov/pubs/trec29/papers/udel_wang_zheng.P.pdf.

Winstead Xingran Zhu. Hotspot detection for automatic podcast trailer generation. Master's thesis, Uppsala University, Department of Linguistics and Philology, 2021.