When Should I Listen to Learn About X? - Podcast Segment Retrieval Based on a Natural Language Queries

Fredrik Segerhammar
fseg@kth.se

Mariya Lazarova
lazarova@kth.se

Tianzong Wang
tianzong@kth.se

Supervisor: Judith Bütepage

Abstract

Locating the best matching paragraph in a document given a search query is a very well studied problem. However, for podcast data the problem is newer and there is not much research done on it. We attempt to retrieve the best jump-in point for relevant segments of podcast episodes given arbitrary user search queries, using the dataset provided in the TREC 2020 Podcasts Track [1]. We propose two methods, one based traditional statistical methods utilizaing TF-IDF and Okapi BM25, and another Sentence-Transformer based deep learning embedding method, to target the Ad-hoc Segment Retrieval task. We provide experimental studies on both methods. For the statistical model we explored different hyperparameters, and for the deep learning method, we explored factors like pre-trained model choice. We test our model performance on a randomly selected 10% subset of the original dataset and provide the evaluation results. We compare the evaluations between the two models, and discuss the advantages and drawbacks of each approach. We also included ethical and sustainability discussions. The code is available under this[1] repository.

1 Introduction

Podcasts are a relatively new form of audio media, whose content is spoken language with different levels of formality. Podcasts are usually episodic, keeping the style and the general topic of the podcast, but ranging the specific topic for each episode. This media format has gained a lot of popularity in recent years, which drives more and more content creation within the field. As the medium grows, there is bigger need for understanding its content. Of course, this is not an easy task due to the format of the data. Podcasts are very often not scripted, so transcribing them is a difficult task, and even when this is done, the transcripts are of sub-optimal quality. Moreover, there is often more than one person speaking in the same episode, so transcripts will be very different from the usual sequential text documents. Due to this fact, it might be difficult for an algorithm to infer the context. In comparison to articles, which are often well written and with well-formatted language, podcasts usually follow a more lax style, linguistically, where words may be repeated several times for example, or several people may speak at the same time. They also usually tend to go on for up to one or two hours, where certain subjects may be revisited. For these reasons, finding the best starting segment is a challenging task.

The problem of finding not only the most relevant podcast and episode, but also the most relevant segment, is important. Frequently, users listen to a podcast in order to learn about a specific topic, and starting always from the beginning of an episode could give them the wrong impression of the

[1] https://github.com/tianzongw/Podcast-Segment-Retrieval-Spotify

episode's content and they might stop listening. Moreover, they might never end up listening to the most relevant episode, because the episode description is misleading or they have not been exposed to the creator. To try to remedy this, the problem of finding the most relevant podcast segment, based on a user query and description is defined.

In this study, we present two conceptually different methods for the task of relevant podcast segment retrieval - one method, utilizing the more traditional statistical information retrieval methods, and one, utilizing the emerging deep learning methods, and compares them (Section 4). Before this, we present relevant ideas (Section 2) and the data we are working with (Section 3). We also present a couple of experiments (Section 5) analyzing different parameters of the models. The test set results can be found in Section 6. At the end, we present ideas for future work (Section 8), ethical and sustainability discussion (Section 7) as well as some ideas for solving our problem that were unsuccessful (Appendix).

2 Related Work

Here, we will examine some different methods that are commonly used in query retrieval for NLP.

2.1 Classical Statistical Methods

There are many different statistical methods in information retrieval, that rely on token counts withing the documents and the search query, to identify which are the important tokens, and based on that rank the documents in the corpus based on the search query. Two such approaches are TF-IDF and BM25.

TF-IDF

TF-IDF, short for term frequency-inverse document frequency, is a metric that is used to determine the relative importance of a word to a document within a corpus of documents D. As the name suggests, TF-IDF is a product of two statistics - the term frequency tf and the inverse document frequency idf. The vanilla version of tf is the raw count of occurrences of the w in d - $f_{w,d}$. A more advanced version of tf is the augmented frequency defined in equation 1, which takes into account the fact that the documents in the corpus are of different length. For idf there are also many different formulations, but the simple version in equation 2 is often good enough. Taking these formulations of tf and idf, we arive at the $tfidf(w,d)$ metric, shown in equation 3.

$$tf(w,d) = 0.5 + 0.5\frac{f_{w,d}}{\max\{f_{t,d} : \text{for } t \text{ in } d\}} \tag{1}$$

$$idf(w,D) = \log_2 \frac{|D|}{|\{d \in D : t \in d\}|} \tag{2}$$

$$tfidf(w,d) = tf(w,d)idf(w,D) \tag{3}$$

For a word w and document d, the $tf - idf(w,d)$ will be high if $idf(w,D)$ is small, or if $tf(w,d)$ is very high. Intuitively, this means that in order for a word to be important for a document, it has to appear many time in the said document, or it has to appear in very few documents.

Okapi BM25

The Okapi BM25, coming from best matching, is a bag-of-words retrieval function, which ranks the documents of a corpus with respect to a user query. It also utilizes the term frequency and inverse document frequency metrics, but with a slightly different definition. Here, $tf(w,d)$ is the vanilla $f(w,d)$ for a a term w and a document d. However, the definition of the idf is more involved than and can be seen in equation 4. Putting it together, the scoring function is defined in equation 5. In this definition, we can see that each term in the query contributes with both it's term frequency within a document and it's document frequency.

$$idf(w,D) = \ln \frac{|D| - n(w) + 0.5}{n(w) + 0.5} + 1, \text{ where } n(w) = |\{d \in D : t \in d\}| \tag{4}$$

$$score(d,q) = \sum_{t \in q} idf(t,D) \frac{f(t,d)(k_1+1)}{f(t,d) + k_1(1 - b + b\frac{|d|}{avg\_dl})} \tag{5}$$

where k_1 and b and parameters in the ranges $[1.2, 2.0]$ and $[0, 1]$ respectively, usually chosen to be 1.5 and 0,75, and avg_dl is the average document length in the corpus.

Once the scores for all documents based on the query are computed, the most relevant documents are the ones with highest score. One could retrieve the n most relevant documents to the query by outputting the n documents with highest scores.

2.2 Deep Learning Methods

Recent advances to extract features from sentences/segments representation focus on generating semantic meaningful embeddings from pre-trained deep learning models. In this section We will review some of the techniques and divide them into supervised and self-supervised classes.

Self-Supervised

Self-supervised methods are generally based on the assumption that adjacent sentence share similar meaning. Inspired by this, **SkipThoughts** [2] proposed the famous Sen2Vec model to convert text data into 4800 dimensional vector representations. For each sentence, it utilizes a encoder-decoder structure to minimize the log-probability sum of the decoder-generated previous and next sentence. Another similar model, **QuickThoughts** [3], instead approaches the prediction of adjacent sentences with a discriminative manner. For each input sentence, after encoding it with the encoder, it then tries to choose the correct target sentence from a set of candidate sentences. In this way the model can choose to ignore irrelevant parts of the target sentence to update its weight. These methods, however, are flexible in terms of the choice of encoders and decoders, but did not incorporate the robustness of pre-trained BERT models. A more recent method, **Info-Sentence BERT** (IS-BERT) [4], applies a set of CNNs with different window size on top of BERT sentence embeddings as local representations. Then it applies average pooling to the last layer of BERT as the global representation. The objective is to maximize the mutual information behind local and global representations.

Supervised

Supervised methods utilize labeled data to minimize the gap between similar sentence representations. **Universal Sentence Encoder** (USE) [5] trains two different encoders, one based on Transformer [6] architecture and another is based on Deep Averaging Network (DAN) [7]. Their outputs are then combined and provided to a task specific DNN. Another method [8] proposed generating universal sentence embeddings by training on Natural Language Inference dataset, from which Sentence-Transformers fine tuned their parameters as well. Another method to mention, **BERT Reranking** [9], treated as a baseline model for deep learning approach on our task. BERT Reranking combines standard mechanism, such as BM25, to first find possible relevant documents and then re-rank the candidate passages by how relevant they are to the user query with BERT.

3 Data

As stated above, podcasts are an emerging media with some specifics of the content and the format. Due to its novelty and specificity, there are almost no podcast specific datasets. Recently, Spotify in collaboration with Text Retrieval Conference released a 100k-episode dataset containing both audio files and auto-transcribed transcripts [1]. All episodes are in English and cover a wide range of topics, lengths, level of formality, number of speakers, and whether it is scripted on not. Due to its versatility, this dataset will be a big step-up for the speech and NLP research on podcasts.

In our work, we will be using the structured transcriptions from a randomly sampled 10% portion of The Spotify Podcast Dataset [1]. Each podcast episode is stored in a json file and is divided into different segments of varying lengths. Each segment is then separated on a word level. The start and end time for each word, as well as the overall confidence of the segment trascription, is provided as well. For the user queries, we will be using the user queries for the TREC challenge [2]. Each query contains an ID, a keyword description and a textual description. The dataset is under Non-disclosure agreement, and access can be given upon request.

4 Methodology

In this section, we will present the two different approaches that we propose for solving the problem of relevant podcast segment retrieval.

4.1 Statistical Method

For this more conventional model, we decided to make a 3-step model - data cleansing and tokenization, episode filtering, and segment filtering.

Data preparation For the statistical method, we first extract all 30sec segments and the full episodes transcripts. We then remove all stop words, punctuation, and lemmatize. After this we turn the text into tokens, using both unigrams and bigrams. In section 6, we will talk about the effect of filtering the digits or adding trigrams.

Episode filtering One of our main ideas was to split the search into coarse and fine search. Our idea for the coarse search if to use the tf-idf metric on a episode level. The way we do this is by summing the tf-idf score for each token of the query for each podcast episode - equation 6, and returning the n most relevant podcast episodes. We have attempted training with different values of n, which will be covered in section 6, and our choice was $n = 20$.

$$score(q,d) = \sum_{t \in q} tfidf(t,d) \tag{6}$$

Segment search After we have the 20 relevant episodes, we take all 60sec long segments, each starting 30sec after the previous one. Once we have our new corpus of 60-second-long segments, we perform the Okapi BM25 on each segment for a query, and output the best matching segments. For testing, we ar outputting 5 segments, however, for training, we take the best 10 segments in order to analyze the model's performance.

4.2 Deep learning approach

BERT

BERT [10] stands for Bidirectional Encoder Representations from Transformers. There are two steps in BERT, pre-training and fine-tuning. Compared with the other then state-of-the-art language representation models, for example, the Generative Pre-trained Transformer (OpenAI GPT) [11], where each token can only attend to its previous tokens in the self-attention layers, BERT fuses both direction context in pre-training by masking 15% of the sentence and force the model to predict the original token. It is also trained to understands sentence relationships on binarily paired sentences with Next Sentence Prediction. In this way, BERT is able to learn the contextual relationship between both words and sentences level. This pre-trained model can then be fine-tuned to a variety of NLP tasks by adding another layer to plug in the task-specific inputs and outputs.

Sentence-Transformers

A problem of BERT is that it requires passing both sentences into the network. In our case, to find the most similar pair would require $O(N^2)$ computations. SentenceTransformers, or S-BERT, is an example of a fine-tuned BERT model. S-BERT extends the original BERT architecture to tasks including large-scale semantics similarity comparison, clustering and information retrieval via semantic search [12], by adding a pooling operation to BERT output layer, and feeding the resulting fixed-length embeddings to a siamese and triplet networks network to update the weights. This approach is trained on the SNLI [13] and Multi-Genre NLI [14] datasets on different task-oriented objective functions (classification, regression and

[2] https://podcastsdataset.byspotify.com/

triplet), such that the produced sentence embeddings also incorporate semantic information in a much more computationally efficient manner. This can then be used to map whether input sequences are semantically close or not, with metrics like cosine similarity, or Manhatten distance.

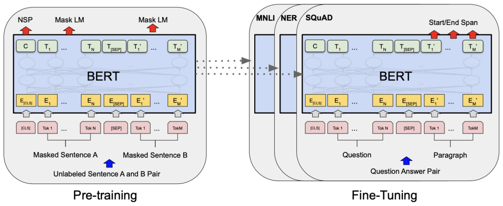


Figure 1: BERT [10]

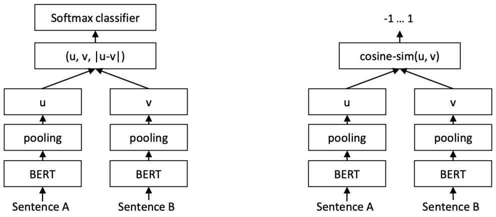


Figure 2: S-BERT [12]

Similarity Matching

In our case, we first extracted the topic descriptions and segments texts from the raw dataset. Then we use Sentence-Transformers models to create fixed length embeddings (768 base, 1024 large). We then rank them based on their cosine similarity score with the topics and kept top-K matched segments. They are finally forward and backward smoothed to create roughly 60 seconds results.

5 Experiments

5.1 Statistical Method

Influence of the ngram range in the statistical method

The first experiment we conducted on the statistical method is on the maximum number of words in an ngram. We tried with n = 1, 2, and 3 (only unigrams, unigrams and bigrams, and uni-, bi-, and trigrams respectively). In the case n=1, our model was not considering some important word combinations like *New York*, *Atlantic Ocean*, *Anna Sorokina*, etc. For many of these word combinations this is not a big problem - *Atlantic* on its own would mean the *Atlantic Ocean*. However, for cases like *New York*, the algorithm would match many documents that have the word *new* in them, which will not be a good match for a query searching for something like *New York*. Most of these issue are solved when adding bigrams. It is important to note though that even though there are bigrams present, sometimes the unigrams have a greater effect - for example for the query *anna delvey* the model returned a lot of results where different *Anna's* were discussed. Generally speaking, the quality of the matching segments with n=2 was significantly better than with n=1. With n=3, however, at a lot higher computational cost we did not notice a significant improvement. Our explanation for this is that the trigrams are very specific, and it is entirely possible that no trigrams from the query appear in the corpus.

Based on these observations, we chose n=2 as the optimal parameter.

Influence of the number of filtered documents

The next experiment we did concerns the number of filtered episodes at the coarse search step. We tried with 20, 60, and 100 filtered episodes. We expected 60 to be a good middle ground between filtering too few to capture the best episode and filtering too many to make the model too computationally expensive. However, we were surprised to find out that 20 episodes yielded better results overall. For most of the training topics, the resulting segments were almost the same, which led us to believe that the extra number of episodes did not contain relevant enough segments on average in this environment. For topic 8 - "facebook stock prediction" in particular, filtering only 20 episodes actually improved the quality of the best segments, as the filtered episodes have been well on topic. For this query when we had 60 filtered episodes, our fine search was focusing too much on words like *prediction*, so the output segments were not on topic. We believe this is because of the fact that TF-IDF is a bag-of-words model, where the episodes containing the most amount of relevant words to the query are likely to be regarded as the most relevant episodes. Essentially forcing the BM25 model to only work with what are likely to be more relevant episodes, also sets BM25 up for success. Based on these observations, we set this parameter to 20.

Influence of filtering out the digits in the pre-processing step

Initially, we were planning to remove all numbers and digits from the training episodes and queries in the pre-processing step. However, we wanted to experiment with numbers not being filtered out, because in some of the query description there are for example dates or key terms containing numbers. For many of the training topics, we did not notice any difference, as there were no numbers present in the query. In topic 8 we did notice improvement when the digits and numbers were present. Because of this, we decided to keep the digits and numbers in our method.

5.2 Sentence-Transformers

For Sentence-Transformer based models, we conducted the experiments regarding three factors: the size of searching space, the quality of matching, number of top matches, as well as the choice of pre-trained BERT models. Searching space refers to the number of randomly selected non-relevant episodes. They are combined with the 100 relevant episodes to form the searching pool. Top-K matching means how many resulting segments we kept yo calculate the accuracy. The match level refers to the given labels, where 3+ was the best, 2+ as good and 1+ as relevant. Base-model choice, in our case, we used the base and large version of BERT and RoBERTa pre-trained models. RoBERTa is an optimized version of BERT with improved training methodology and 10 times the training data. The base and large here refers to the difference in sizes, or more specifically, in number of stacked encoders, the hidden size and the number of heads in the Attention layers.

The results are shown as accuracy. Note that we have 8 training topics and about 100 positive labels. To count as a correct match, we first extract the starting times from the top-k resulting segments. Then we collect the given matching starting time and apply -/+30s to transform them into 60s target intervals. For each topic, if any top-k resulting start time sits in one of the target interval, we count it as a correct match. The summarized results are shown as tables below. The results are averaged from two random trials, i.e., randomly sampling searching space from the whole dataset. Here the matching level best.

Top 1 match

Settings		Model			
Search Space	Match	BERT Base	BERT Large	RoBERTa Base	RoBERTa Large
200	best	0.25	0.375	0.5	0.5
	good	0.375	0.5	0.5	0.625
	relevant	0.375	0.625	0.5	0.75
500	best	0.125	0.375	0.5	0.375
	good	0.375	0.5	0.5	0.625
	relevant	0.25	0.5	0.5	0.75
10,000	best	0.125	0.25	0.375	0.375
	good	0.25	0.375	0.375	0.625
	relevant	0.25	0.5	0.5	0.75
20,000	best	0.125	0.375	0.375	0.375
	good	0.25	0.375	0.25	0.625
	relevant	0.25	0.5	0.375	0.625
50,000	best	x	x	0.125	x
	good	x	x	0.125	x
	relevant	x	x	0.375	x

Top 3 match

Settings		Model			
Search Space	Match	BERT Base	BERT Large	RoBERTa Base	RoBERTa Large
200	best	0.375	0.625	0.625	0.875
	good	0.5	0.625	0.625	0.75
	relevant	0.75	0.75	0.75	0.875
500	best	0.25	0.625	0.5	0.75
	good	0.5	0.625	0.625	0.75
	relevant	0.625	0.75	0.625	0.875
10,000	best	0.25	0.625	0.375	0.75
	good	0.5	0.625	0.625	0.625
	relevant	0.5	0.75	0.625	0.875
20,000	best	0.25	0.625	0.375	0.75
	good	0.5	0.625	0.5	0.625
	relevant	0.375	0.75	0.375	0.75
50,000	best	x	x	0.125	x
	good	x	x	0.25	x
	relevant	x	x	0.375	x

Top 5 match

Settings		Model			
Search Space	Match	BERT Base	BERT Large	RoBERTa Base	RoBERTa Large
200	best	0.5	0.75	0.75	0.875
	good	0.625	0.625	0.625	0.75
	relevant	0.875	0.75	0.875	1.0
500	best	0.5	0.75	0.625	0.75
	good	0.625	0.625	0.625	0.75
	relevant	0.75	0.75	0.75	0.875
10,000	best	0.5	0.625	0.625	0.75
	good	0.625	0.625	0.625	0.75
	relevant	0.75	0.75	0.75	0.875
20,000	best	0.5	0.625	0.625	0.75
	good	0.5	0.625	0.625	0.75
	relevant	0.5	0.75	0.625	0.875
50,000	best	x	x	0.375	x
	good	x	x	0.5	x
	relevant	x	x	0.5	x

6 Results

In this section we will present the results we have achieved on the test set of the TREC challenge, which consist of fifty different in style user queries and descriptions. We will also present our experiments and the results from them. The scoring system is detailed as follows:

- 3: The segment conveys highly relevant information, is an ideal entry point for a human listener, and is fully on topic. An example would be a segment that begins at or very close to the start of a discussion on the topic, immediately signaling relevance and context to the user.
- 2: The segment conveys highly-to-somewhat relevant information, is a good entry point for a human listener, and is fully to mostly on topic. An example would be a segment that is a few minutes "off" in terms of position, so that while it is relevant to the user's information need, they might have preferred to start two minutes earlier or later.
- 1: The segment conveys somewhat relevant information, but is a sub-par entry point for a human listener and may not be fully on topic. Examples would be segments that switch from non-relevant to relevant (so that the listener is not able to immediately understand the relevance of the segment), segments that start well into a discussion without providing enough context for understanding, etc.

- 0: The segment is not relevant.

Statistical method

Here, we are presenting the average topic score, which is the average of the scores of the best matching segments for the topic.
The statistical model achieves different average scores depending on the topic as can be seen in figure 3. Our model works very well for 5 of the test topics. By examining the topics in this well performing category, for example topic 9 - "trump call ukrainian president", we can see that these are topics that have attracted a lot of media attention, and have probably been covered by many professional journalism podcasts. If we look in to the group with average score between 1 and 2, we can see that there are often some relevant names or words, but the topics are not so popular within the Western English-speaking world - for example topic 12 - "imran khan career", or topic 43 - "hong kong protest". We can also see, that the model has not been able to find even one slightly relevant segment for nine of the topics. An example from this group is topic 24 - "france yellow vest protests".

The average scores on each topic can be seen on figure 3

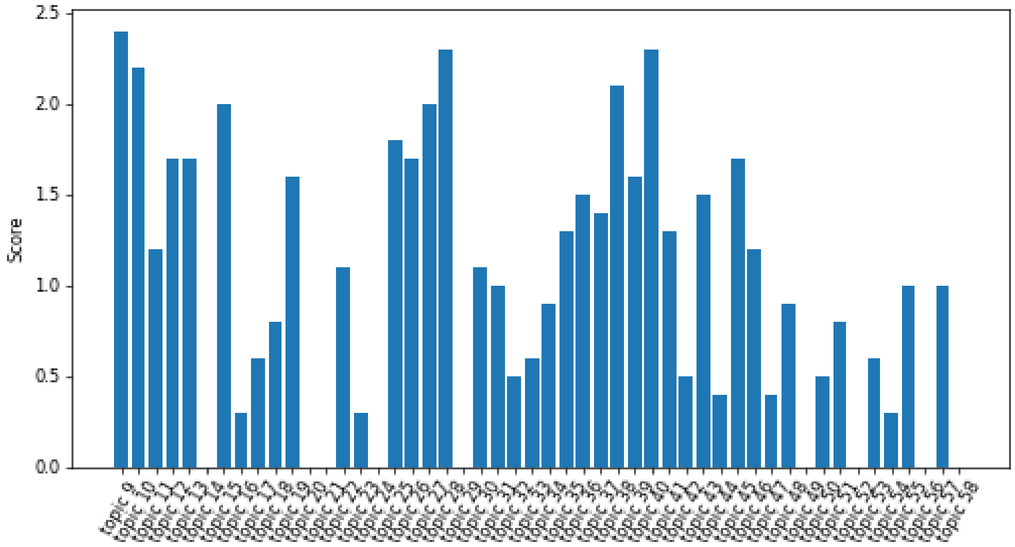
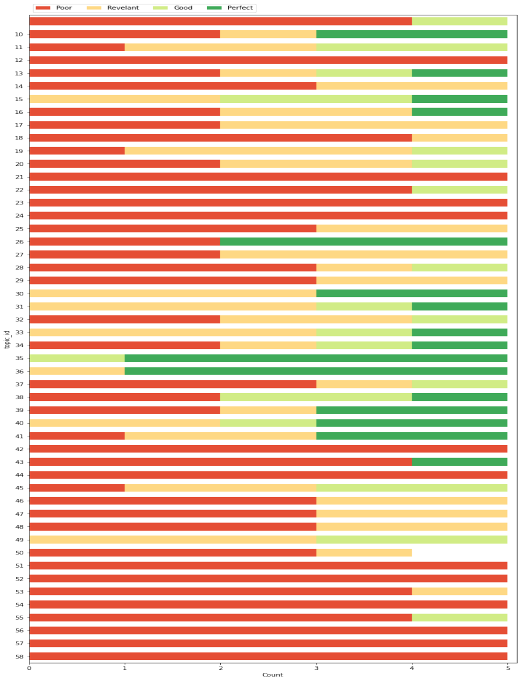


Figure 3: Results from the statistical method

Sentence-Transformers

Below is an example of the score distribution between Sentence-Transformers and statistical method from the same grader. The S-BERT model used here is RoBERTa Large. As is shown on the plot, Keyword-based statistical method performed better on most of the test topics, with an average score of 1.269, whereas 0.816 for Sentence-Transformers. Our opponent group achieved an average score of 0.168 for their statistical method, and 0.225 for their deep learning method with Wiki-Embedding [15].



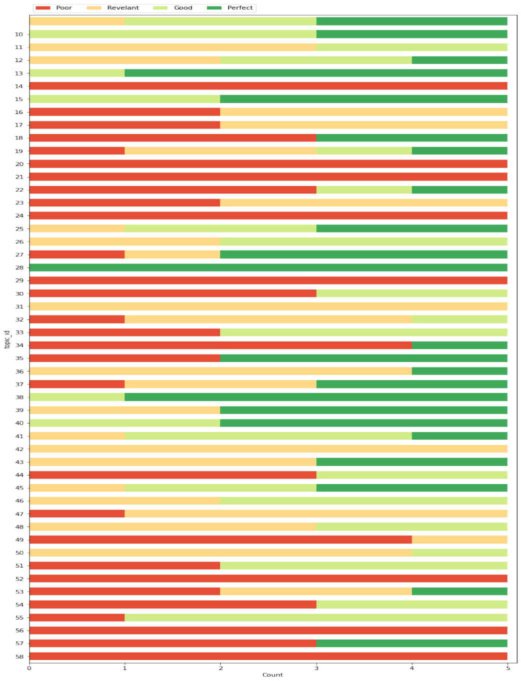


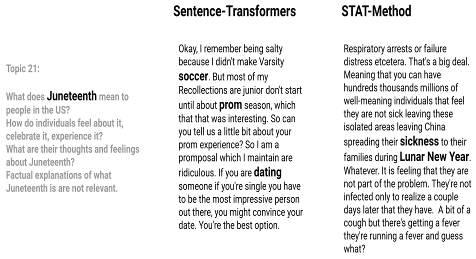
Figure 4: Sentence-Transformers test result

Figure 5: Statistical Method test result

Here, poor, relevant, good and perfect refers to 0, 1, 2, 3 in the previously mentioned grading scheme.

As shown below, there exist common 'difficult' topics, for example, topic 21 juneteenth. A quick glance showed that the result segments is either mapped to the wrong keyword for statistical method, and random search for Sentence-Transformers. This may result from the fact that the searching space did not contain the target episode.

For topics where statistical model performed much better than Sentence-Transformers, for example, topic 28 yo-yo dieting, statistical method performed a good catch on the keyword 'yo-yo', while Sentence-Transformers either mapped to other method, or random segments. This may come from the fact that 'yo-yo' dieting is an out-of-vocabulary for Sentence-Transformers, and Sentence-Transformers does not generalize well on out-of-distribution (OOD) domains.



Topic 21:

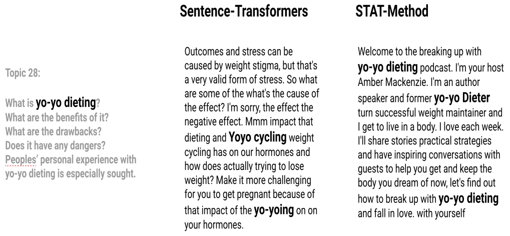
What does **Juneteenth** mean to people in the US?
How do individuals feel about it, celebrate it, experience it?
What are their thoughts and feelings about Juneteenth?
Factual explanations of what Juneteenth is are not relevant.

Sentence-Transformers

Okay, I remember being salty because I didn't make Varsity **soccer**. But most of my Recollections are junior don't start until about **prom** season, which that that was interesting. So can you tell us a little bit about your prom experience? So I am a promposal which I maintain are ridiculous. If you are **dating** someone if you're single you have to be the most impressive person out there, you might convince your date. You're the best option.

STAT-Method

Respiratory arrests or failure distress etcetera. That's a big deal. Meaning that you can have hundreds thousands millions of well-meaning individuals that feel they are not sick leaving these isolated areas leaving China spreading their **sickness** to their families during **Lunar New Year**. Whatever. It is feeling that they are not part of the problem. They're not infected only to realize a couple days later that they have. A bit of a cough but there's getting a fever they're running a fever and guess what?



Topic 28:

What is **yo-yo dieting**?
What are the benefits of it?
What are the drawbacks?
Does it have any dangers?
Peoples' personal experience with yo-yo dieting is especially sought.

Sentence-Transformers

Outcomes and stress can be caused by weight stigma, but that's a very valid form of stress. So what are some of the what's the cause of the effect? I'm sorry, the effect the negative effect. Mmm impact that dieting and **Yoyo cycling** weight cycling has on our hormones and how does actually trying to lose weight? Make it more challenging for you to get pregnant because of that impact of the **yo-yoing** on on your hormones.

STAT-Method

Welcome to the breaking up with **yo-yo dieting** podcast. I'm your host Amber Mackenzie. I'm an author speaker and former **yo-yo Dieter** turn successful weight maintainer and I get to live in a body. I love each week. I'll share stories practical strategies and have inspiring conversations with guests to help you get and keep the body you dream of now, let's find out how to break up with **yo-yo dieting** and fall in love. with yourself

Figure 6: topic 21 result

Figure 7: topic 28 result

7 Ethical and Sustainability Considerations

While the reported results here are satisfactory, there are some other ethical and sustainability considerations we need to take.

First of all, the task we are concerned with is finding the most relevant podcast segments. In this setting we are not considering fair suggestions. Therefore, we are not suggesting more episodes from podcasts by women, LGBT+, people of color, or other groups that could be poorly represented in the dataset. There are two main reasons for this. First of all, our algorithms do not account for the identitities of podcast participators. Second of all, the dataset does not contain any of this information, so this makes the task of fair suggestions harder. If the podcasts within the dataset were to be labeled with this information, it is possible that we could more easily suggest podcasts more fairly.

However, although there seems to be no partiality or inclination in programming both our algorithms as well as the algorithms used to procure the data, human bias and subjective tendency are omnipresent in steps like training data collection and feature engineering, which raises a concern about the neutrality of the pre-trained model based deep learning approach. In the recently held **AI DEBATE 2**[3], Margaret Mitchell mentioned the concept of 'bias laundering'. It might be interesting to test if our model is biased towards certain groups or values, for example, genders or political views, from semantically equally relevant segments.

The second note we would like to address, is that we achieve better results with a traditional statistical method. This is great, as this type of method is typically easier to train in terms of computing resources and energy consumption. While in our case, the deep learning approach took less time to give suggestions, it is important to note that this is because it is using pre-trained weights. If we were to train from scratch, it would take a lot more power and training time.

8 Conclusion

8.1 Statistical Method

We find that, for this task, using an ngram with $n = 3$ will have a detrimental effect on the results. It is unclear whether this is task-specific, or if in a real world setting, the results may be different. However, for the queries used in the training and test data, an $n = 2$ proved to be generally optimal, although $n = 1$ sometimes proved to be superior if the key term were a single word. Some sort of filtering to restrict BM25 to the most relevant documents also proved superior for this regime, however it is possible that with enough data the top results could contain enough relevant podcasts so that filtering is not needed. For our assessment we used 10000 podcast episodes. In reality, far more will be used in an application, and then filtering might not be needed.

8.2 Sentence-Transformers

As shown from the experiments result and the comparison. We can conclude that:

1. Model choice is essential for a successful matching. The number of parameters determines if the model can generate semantically meaningful sentence embeddings.
2. Bigger models, for example, RoBERTa Large based models, are more robust in terms of searching space and matching level, and were thus chosen as the testing model.
3. Top-1 match is not robust enough in terms of cosine similarity comparison. To ensure the quality of a matching, we shall use at least top-5 matching.
4. Sentence-Transformers do not generalize as well on out-of-vocabulary terms.

8.3 Comparison

The models seem to achieve similar performance overall. The statistical model is able to extract segments which, most of the time, at least contain relevant key words even when the segments are not relevant. It remains unclear whether there is sufficient data to assess the algorithms for all the queries, or whether relevant podcasts simply did not exist for some queries. While this provides a challenge in assessing the algorithms, it should not affect the comparison, and could in fact provide interesting data

[3] https://montrealartificialintelligence.com/aidebate2.html

9 Future Work

9.1 Statistical Method

For the future, it would be interesting to study different statistical regimes to find a more optimal one than the one we developed here.

There are a lot of different version of BM25 - in [16], the authors propose different improvements on the BM25. It will be interesting to explore how the improved BM25 will perform in such setting. Another avenue to take could be with topic modelling. As we mention in the Appendix, we have attempted topic modelling without success. However, we believe there is potential in topic modelling as an approach for this task. On way it could be applied is in the coarse search task in order to choose the episodes that resemble the query the most in terms of topic representations. Another possible way to utilize it could be my making a hybrid fine search using both topics and word counts.

9.2 Sentence-Transformer

- Quality of auto-transcribed podcast segments
  The results from Wiki-Embedder showed that the quality of the transcriptions is not ideal. There exists several mis-transcribed and mis-spelled words, as well as grammar mistakes within the segments. This could have an impact on either embedding or similarity matching. In the future, a pre-correction linguistic model can be applied before feeding the segments into the model.
- Fine-tuning Sentence-Transformer model with semi-supervised learning
  Sentence-Transformers are famous for their robustness in tasks including semantic textual similarity, semantic search, or paraphrase mining. However, it was fine-tuned on Natural Language Inference datasets, which may result in bias on this task. The main issue for our dataset is the number of positive labels. There are in total 10,000 podcast episodes and 3.4 million segments, whereas only 100 positive labels covering 9 topics only. It is thus difficult to further fine-tune the model downstream on our dataset. Both the experiments results and the score from the opponent group showed that among the extracted segments, there exist good matched pairs. This fact entails that we might be able to fine-tune the model with its own predictions in a semi-supervised manner, like in the recent DivideMix [17].
- Contrastive Tension
  A paper under review for ICLR 2021 [18] proposes the Contrastive Tension method. It first performed a layer-wise study of the semantic quality of the sentence representations in a series of Transformer architectures and noticed that the semantic similarity decreases as layers go deep. This is not beneficial for learning semantic sentence representations. Thus it raised the concern that it is unclear whether the impressive improvements of S-BERT are mainly attributed to the NLI task itself, or the model really captures the semantic information. It then proposed a self-supervised approach meant to encourage the model to retain a semantically distinguishable sentence representation by forcing the model to generate similar embeddings for same sentences, without any task-specific objectives. It will be interesting to test this method on our task as well.

9.3 Other possible avenues to follow

It may also be interesting to test the combination of the statistical method as an episode pre-filter and BERT embeddings as a segment ranker, to see how well they tackle this task together.

References

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

[2] Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S. Zemel, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. Skip-thought vectors, 2015.

[3] Lajanugen Logeswaran and Honglak Lee. An efficient framework for learning sentence representations, 2018.

[4] Yan Zhang, Ruidan He, Zuozhu Liu, Kwan Hui Lim, and Lidong Bing. An unsupervised sentence embedding method bymutual information maximization, 2020.

[5] Daniel Cer, Yinfei Yang, Sheng yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. Universal sentence encoder, 2018.

[6] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.

[7] Mohit Iyyer, Varun Manjunatha, Jordan Boyd-Graber, and Hal Daumé III. Deep unordered composition rivals syntactic methods for text classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1681–1691, Beijing, China, July 2015. Association for Computational Linguistics.

[8] Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes. Supervised learning of universal sentence representations from natural language inference data, 2018.

[9] Rodrigo Nogueira and Kyunghyun Cho. Passage re-ranking with bert, 2020.

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[11] A. Radford. Improving language understanding by generative pre-training. 2018.

[12] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks, 2019.

[13] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2015.

[14] Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics, 2018.

[15] Ehsan Sherkat and Evangelos Milios. Vector embedding of wikipedia concepts and entities, 2017.

[16] Andrew Trotman, Antti Puurula, and Blake Burgess. Improvements to bm25 and language models examined.

[17] Junnan Li, Richard Socher, and Steven C. H. Hoi. Dividemix: Learning with noisy labels as semi-supervised learning, 2020.

[18] Anonymous. Deep representational re-tuning using contrastive tension. In *Submitted to International Conference on Learning Representations*, 2021. under review.

Appendix - Tried and Failed

BM25 on BM25

We attempted to both use BM25 to find the optimal podcast, and also the optimal segment. However, while this did work, there are differences in how TF-IDF and BM25 function that affect the overall result. TF-IDF rewards documents in which terms occur more frequently. BM25, while fairly similar, will not reward texts as much where term frequency is very high. We found that by first locating the podcasts which contained the most key words, and then using BM25 to find the optimal segments proved to be a better strategy.

Latent Dirichlet Allocation

Another alternative to using TF-IDF to reduce the number of podcasts for segment selection is to use Latent Dirichlet Allocation (LDA) to group podcasts into a set of categories which can be set from the queries. However, this is not easily applied to any topic a person could be interested in, and the number of podcasts in a given topic is still too high. Furthermore, combining this method with the TF-IDF & BM25 method as a first stage of filtering episodes does not provide much in terms of value either. Therefore, we quickly moved away from this idea.