DCU-ADAPT at the TREC 2020 Podcasts Track

Yasufumi Moriya, Gareth J. F. Jones

ADAPT Centre, School of Computing, Dublin City University, Dublin, Ireland

{yasufumi.moriya,gareth.jones}@adaptcentre.ie

Abstract

We describe DCU-ADAPT's participation in the TREC 2020 Podcasts Track. We participated in the ad-hoc segment retrieval task. The goal of the task was to search for fixed-length segments from a large archive of podcasts which contain good jump-in points to relevant content for a given query topic. The challenge of retrieving relevant segments with good jump-in points at high rank is made more difficult by the presence of transcription errors in transcripts created using automatic speech recognition. We investigated three query expansion techniques designed overcome this issue. Our first approach was to extract nouns and named entities from the query description provided with each query and to add them to the corresponding query. Our second approach was to retrieve documents for the query using a commercial online web search engine and add selected words from the web documents to the query. Our final approach was to select words to expand the query using a pseudo-relevance feedback method and WordNet. Combining the above approaches for query expansion, we achieved a normalised discounted cumulative gain (nDCG) value of 0.586.

1. Introduction

The growing multimedia archives available on the Internet are increasing the demand for effective tools to enable efficient engagement with this content, including tools for search and summarisation. While tools for search of online text document archives are well advanced, the functionality of tools for spoken content remains very limited. For example, enabling search only using titles and metadata, users may though be interested in a particular relevant region of a spoken document. More sophisticated functionality of the search system, to enable such regions to be identified requires what is actually said to be determined and for this transcript to be used within an information access system. The ad-hoc segment retrieval task of the TREC 2020 Podcasts Track provided a challenge to promote work into facilitating effective search of errorful automatic speech recognition (ASR) created podcast transcripts. Participants were asked to find good "jump-in" points relevant for a given query rather than to retrieve whole podcasts relevant to a query.

Standard spoken content retrieval systems use speech transcripts created by an ASR system with a conventional text retrieval algorithm operating on the speech transcripts [1]. One of the issues of this approach is caused by the errors contained in the speech transcripts. Even when query words are spoken in a document, a search system may be unable to identify a relevant document due to the absence of the query word from the ASR created textual representation of the document. An example from the data provided for the Podcasts is for the query "Daniel Ek interview". The provided ASR transcript system mis-transcribed the family name "Ek" to "eck" or "ech", and there was no instance of "Ek" in the provided collection of speech transcripts. Another issue not only limited to spoken search but also to textual search is the possibility of semantic mismatch between query words and documents. When a given search query is broader than the corresponding expressions used in relevant documents, the retrieved documents may not satisfy user information needs. An example found in the podcasts data is the query "story about riding a bird". Despite the query word "bird", target documents relevant to this query contain a story about riding a "seagull".

In order to overcome these issues, we investigated query expansion for ad-hoc podcast retrieval using three different resources. The goal of query expansion is to add terms to the original query which are useful for discovering relevant documents. The first resource we found to be useful was the query descriptions provided with the task queries. These are longer descriptions of the information need. Some terms found in the description can provide more detail to the original queries increasing the likelihood that they will match with relevant content. The second resource we investigated was an external web search engine. We retrieved web pages for each of the provided queries, and extracted potentially useful query expansion terms from the web pages. These terms were then added to the original queries. The third resource was WordNet which captures semantic relationships between words. This approach was used to mitigate the semantic mismatch between query words and words contained in relevant documents. Overall, by combining all of the above approaches for query expansion, our best run achieved an nDCG value of 0.586.

2. Task Description

In this section, we outline the TREC 2020 Podcasts Track [2] and the podcasts dataset provided for the track by Spotify [3]. The TREC 2020 Podcasts Track offered two tasks: ad-hoc segment retrieval and summarisation. The goal of the ad-hoc segment retrieval task was to find good starting points of podcast episodes given a user search query. The goal of the summarisation task was to produce a text snippet of given audio and a speech transcript. The DCU-ADAPT team participated in the ad-hoc segment retrieval task.

The podcasts dataset contains approximately 100,000 episodes of podcasts. Participants were provided with automatic transcripts of all of the episodes of podcasts. Along with the automatic transcripts, raw audio files of all of the episodes were available to participants. For the retrieval task, participants were provided with 8 training queries accompanied by a set of relevance judgements for development and assessment of retrieval systems. The provided relevance judgements evaluate relevance of documents on a 0-4 value scale. The submitted runs were evaluated on 50 test queries. The types of queries vary from queries looking for a specific named entity "Daniel Ek interview" to more general queries "gaslighting".

In the ad-hoc segment retrieval task, participants were asked to return a maximum of 1,000 podcast segments for each query. The relevance of these segments was rated based on their relevance to the information need expressed in the query and whether they provided a good "jump-in" point for the given query. The length of each segment was requested by the task organisers to be 2 minutes (120 seconds). The offset of each segment needed to be at multiples of 60 (i.e., 0, 60, 120). The submitted systems were evaluated according to normalised discounted cumulative gain (nDCG). While mean average precision (MAP) is a popular metric for information retrieval systems, nDCG can take account of the order of retrieved documents (i.e., some documents are more relevant and should be ranked higher than less relevant documents).

3. System Description

In this section, we describe our system architecture and the query expansion approaches that we investigated to address for this task, focusing particularly on methods which seek to address transcription errors and semantic mismatches between queries and transcripts as discussed in Section 1.

3.1. Data pre-processing

As the task requires participants to submit 120 seconds segments of podcasts whose offset should be multiples of 60, the provided ASR transcripts were segmented into 2 minute chunks. This means that there was an overlap between one transcript segment and the next one (e.g., segment offset 0 and segment offset 60). Words spanning across two segments (e.g., start time is 59.5 and end time is 60.5) were assigned to the first segment.

3.2. Indexing and weight model

We used PyTerrier to index the data and perform search over the indexed data [4]. We applied the divergence from randomness model using Popper's normalisation (DPH) to the collection to rank podcast segments [5]. The advantage of the DPH model over a popular retrieval model such as BM25 [6] is that the DPH model is parameter-free whereas hyper-parameters need to be tuned to operate alternative models such as BM25. Empirically, we found a small difference between DPH and BM25 models when these models were evaluated on the training queries and judgements provided for the task. The DPH model was applied to the collection with the Bo1 query expansion model [5]. Bo1 is a query expansion model which uses pseudo-relevance feedback to assign weights to each query term. The DPH model ranks podcast segments and the Bo1 model re-assigns a weight to query terms using pseudo-relevance feedback. This was our basic retrieval model, and was also applied with the extended queries which contain both original query terms and additional query terms found by the expansion approaches introduced in the following sub-sections.

3.3. Query Expansion using Additional Resources

As discussed in Section 1, transcription errors contained in podcasts and semantic mismatches between query words and documents can be an obstacle to effective search. To address these issues, we investigated query expansion approaches using query descriptions, a web search engine and WordNet. Query expansion can increase the possibility of documents being found compared to the original queries. The three approaches we employed are illustrated in Figure 1. Each of these has the potential to discover important query terms and can complement each other.

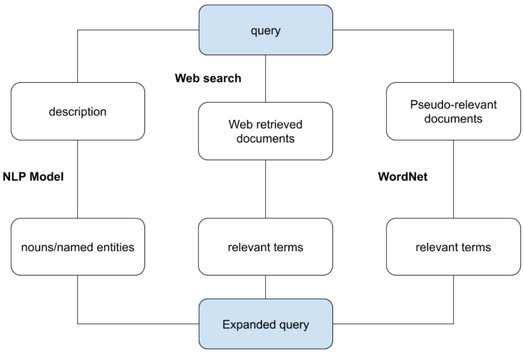

Figure 1: *Query expansion approaches using external resources.*

3.3.1. Query Description

The provided training and test queries are accompanied by a description field giving more details of each information need. We noticed that some of the words included in these descriptions might be useful to match with relevant documents. We applied a part-of-speech tagger and a named entity recogniser to the query descriptions using SpaCy[1]. Identified nouns or/and named entities were added to the original query as additional search terms. While nouns from the query descriptions alone could be useful information for queries (e.g., "Wuhan" for the query "Coronavirus spread"), the named entity recogniser can extract dates and years (e.g., "2019" for the "Coronavirus spread"). It should be noted that all of the nouns and named entities identified by the SpaCy model are added to the original queries. This approach is referred in our experiments to as "nouns" and "entities".

3.3.2. Use of a Web Search Engine

While query descriptions may contain some useful terms for search operations, the number of terms available is limited. Using a public web search engine allows a query expansion approach to find important terms from topically related documents on the Internet. We retrieved 10 web pages for each query using the Google search API. 10 pages were used since this is the stand setting ot the Google search API. Adopting a pseudo relevance approach, we assumed that all of the retrieved 10 pages were relevant to the original query, and calculated Robertson's offer weight for each term $t(i)$ in the 10 web pages. To obtain the offer weight of a term, a relevance weight $rw(i)$ needs to be computed first:

$$rw(i) = \log \frac{(r(i) + 0.5)(N - n(i) - R + r(i) + 0.5)}{(n(i) - r(i) + 0.5)(R - r(i) + 0.5)} \quad (1)$$

where $n(i)$ was the number of podcast segments term $t(i)$ occurs in, N was the total number of podcast segments, $r(i)$ was the number of known relevant documents term $t(i)$ occurs in and R was the number of known relevant documents, in our case the 10 assumed relevant documents retrieved usng the Google search API. The offer weight of a term $t(i)$ was then computed by:

$$ow(i) = r(i) \times rw(i) \quad (2)$$

[1] https://spacy.io

The potential expansion terms were ranked according to the offer weight. For each query, the top 5 terms according to the offer weight were added to the original query. The number of expansion terms was determined by nDCG values on the training queries. This approach is referred in our investigation to as “web search”.

3.3.3. Use of WordNet

The query extension approach using a web search engine is effective when the original query contains specific terms in particular named entities, while it is hard to obtain informative terms from a generic query (e.g., “story about riding a bird”). Our third approach aimed to obtain hypernyms and hyponyms of a query term using pseudo-relevance feedback. A collection of podcast segments was first ranked by the DPH model using Bo1 query expansion. The top 50 ranked documents for each query were assumed to be relevant. Then, a part-of-speech tagger was applied to the queries and to the pseudo relevant documents to retain only nouns. To compute relevance of query nouns and nouns in the pseudo relevant document collection, Wu-Palmer similarity [7] was applied to the two nouns. The Wu-Palmer similarity can measure semantic closeness of two words exploiting the graph structure of WordNet. When the similarity score of a noun exceeds 0.8, the word was assumed to be relevant to a query noun. Finally, the terms semantically relevant to the queries were ranked according to the Robertson offer weight (Section 3.3.2). Terms were added to queries when their weight was higher than 5.0. All of the parameters mentioned for this approach were determined by examining nDCG values on the training queries. This approach is referred to as “WordNet”.

3.4. Run descriptions

We submitted 5 runs combining the query expansion approaches introduced in the previous sections. Combination of two or more query expansion methods was performed by adding all of the expansion terms to the original queries and applying the DPH model with the Bo1 query expansion to the extended queries.

- Run1: nouns and WordNet
- Run2: nouns and entities
- Run3: nouns, entities and WordNet
- Run4: nouns, entities and web search
- Run5: nouns, entities, WordNet and web search

4. Results and analysis

Table 1 shows the results of our submitted 5 runs. Overall, Run 5 was the best submission achieving and nDCG value of 0.586, followed by Run 4 with 0.581. The results also show that query expansion using web search (Run 4 and Run 5) boosted nDCG compared to the other 3 runs. The result overall show the positive impact of using WordNet expansion. This is an interesting result, since query expansion using WordNet is generally not found to be effective for IR. However, out $ow(i)$ term selection method appears to work well in combination with the potential expansion terms found in WordNet, and we will investigate this in more detail in further work. Examining the results for Run1, Run 2 and Run 3 in comparison, it is not clear whether the noun expansion method makes a positive contribution, and further experimentation is needed to better understand these results.

Table 1: *nDCG results of our 5 submitted runs*

Run ID	nDCG
Run 1	0.570
Run 2	0.551
Run 3	0.569
Run 4	0.581
Run 5	**0.586**

Table 2: *nDCG results of 9 query IDs with interesting differences between the submitted runs. Our Run 5 scored the highest of all submitted systems on the first 5 queries. On the bottom 4 queries, adding terms found in the web documents decreased nDCG values by more than 0.1.*

Query ID	Run 1	Run 2	Run 3	Run 4	Run 5
11	0.605	0.605	0.605	0.74	0.74
12	0.765	0.618	0.765	0.638	0.772
18	0.55	0.55	0.554	0.623	0.638
52	0.302	0.302	0.302	0.861	0.861
55	0.45	0.576	0.45	0.845	0.856
16	0.697	0.706	0.711	0.596	0.569
36	0.686	0.686	0.686	0.569	0.569
46	0.903	0.903	0.903	0.778	0.778
56	0.728	0.728	0.728	0.284	0.284

Participants were provided with the summary table of submission runs from all participants. This shows minimum, median and maximum nDCG values of each query. Comparing our best Run 5 to the median values, it turned out that our Run 5 produced an nDCG value higher than the median for 36 out of 50 test queries. Furthermore, Run 5 achieved the highest nDCG values of all the submitted systems for query IDs 11, 12, 18, 52 and 55. These queries are “how to cook turkey”, “Imran Khan career”, “Women in STEM”, “Fauci interview” and “Malcolm X biography”. As can be seen in Table 2, higher nDCG values from Run 4 and Run 5 show that query expansion using the web search engine was beneficial for queries 11, 18, 52 and 55. The query expansion method found terms, for example, “oven” and “gravy” for the topic “how to cook turkey”, “coronavirus” and “Anthony (the first name of Fauci)” for the topic “Fauci interview” and “Mecca” and “Muhammad” for the topic “Malcolm X biography”. Query 12 is an example where Run 1, Run 3 and Run 5 were better than Run 2 and Run 4 indicating that terms found from the WordNet were useful. For the topic “Imran Khan career”, terms such as “job” and “assistance” were discovered as important terms to be added to the query.

The query expansion approach using the web search engine, however, did not always bring benefit to the search system. There are 4 queries whose nDCG value for Run 5 was 0.1 or lower than that of Run 3, indicating that adding terms from the web pages had a negative impact on ranked documents in these cases. These queries are “causes and prevention of wild fires (query 16)”, “Cryptocurrency risks (query 36)”, “Missouri quilt mom (query 46)” and “gaslighting (query 56)”. Despite the lower nDCG score, the terms obtained from web search seem to be valid. For example, the terms such as “bitcoin” and “currencies” were extracted for the query “cryptocurrency risks” and “victim” and “sanity” for the query “gaslighting”.

5. Conclusions

In this paper, we described the DCU-ADAPT team participation in the TREC 2020 Podcasts Track. The Podcasts Track offered an ad-hoc segment retrieval task and a summarisation task. We participated in the ad-hoc segment retrieval task and explored the effects of using external sources for query expansion. The first query expansion method extracted nouns and named entities from query descriptions and adds them to the original queries. The second approach exploited an external web search engine to retrieve relevant documents, and extracted important terms from the web documents. The third approach used pseudo-relevance feedback and extracts relevant terms from WordNet as potential expsnsion terms. Our best run combining all of the above query expansion methods achieved 0.586 nDCG. Our analysis revealed that certain terms discovered by query expansion using the web search engine were particularly effective on ranking podcast segments, although this approach had a negative impact on more generic queries.

6. Acknowledgements

This work was supported by Science Foundation Ireland as part of the ADAPT Centre (Grant 13/RC/2106) at Dublin City University.

7. References

[1] M. Larson and G. J. F. Jones, “Spoken Content Retrieval: A survey of techniques and technologies,” *Foundations and Trends in Information Retrieval*, vol. 4, no. 4-5, pp. 235–422, 2012.

[2] R. Jones, B. Carterette, A. Clifton, M. Eskevich, G. J. F. Jones, J. Karlgren, A. Pappu, S. Reddy, and Y. Yu, “TREC 2020 Podcasts Track Overview,” in *The 29th Text Retrieval Conference (TREC) notebook*. NIST, 2020.

[3] A. Clifton, S. Reddy, Y. Yu, A. Pappu, R. Rezapour, H. Bonab, M. Eskevich, G. J. F. Jones, J. Karlgren, B. Carterette, and R. Jones, “100,000 Podcasts: A Spoken English Document Corpus,” in *In Proceedings of the 28th International Conference on Computational Linguistics (COLING 2020)*, 2020, pp. 5903–5917.

[4] C. Macdonald and N. Tonellotto, “Declarative experimentation in information retrieval using PyTerrier,” in *Proceedings of ICTIR 2020*, 2020.

[5] G. Amati and C. J. Van Rijsbergen, “Probabilistic models of information retrieval based on measuring the divergence from randomness,” *ACM Transactions on Information Systems (TOIS)*, vol. 20, no. 4, pp. 357–389, 2002.

[6] S. Robertson, S. Walker, S. Jones, M. Hancock-Beaulieu, and M. Gatford, “Okapi at TREC-3,” *NIST special publication*, no. 500225, pp. 109–123, 1995.

[7] Z. Wu and M. Palmer, “Verbs semantics and lexical selection,” in *Proceedings of the 32nd Annual Meeting on Association for Computational Linguistics*, 1994, p. 133–138.