Rich Speech Retrieval Using Query Word Filter

Christian Wartena
Univ. of Applied Sciences and Arts Hannover *
Hannover, Germany
christian.wartena@fh-hannover.de

Martha Larson
Delft University of Technology
Delft, the Netherlands
m.a.larson@tudelft.nl

ABSTRACT

Rich Speech Retrieval performance improves when general query-language words are filtered and both speech recognition transcripts and metadata are indexed via BM25F(ields).

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: Content Analysis and Indexing

General Terms

Algorithms, Experimentation

Keywords

Spoken content retrieval, Query word classification

1. INTRODUCTION

Our Rich Speech Retrieval (RSR) approach filters words in the query into two categories and treats each separately. RSR is a known-item task that involves returning a ranked list of jump-in points in response to a user query describing a segment of video in which someone is speaking. The queries are given in two formulations: a long form consisting of a natural language description of what the known item is about (ca. one sentence in length) and a short form consisting of a keyword version of the query as it might be issued to a general-purpose search engine. The video corpus used contains Creative Commons content collected from blip.tv and the spoken channel is a mixture of planned and spontaneous speech. Although visual features might prove helpful for some RSR queries, here, we investigate only the use of ASR-transcripts and metadata. Note that although the know-items targeted in the RSR task correspond to particular speech acts, we did not investigate this aspect here. More details on the RSR task are available in [4].

We conjecture that users queries are a mixture of two distinct types of language: general query language and primary language. General query language is language the users always use when formulating queries for videos during a search session with a general search engine (e.g., *video*, *episode*, *show*). Our conjecture is based on informal observation of user query behavior. It is supported by a user study of podcast search behavior [1] during which subjects reported adding general words such as 'podcast', 'audio' or 'mp3' to queries when looking for podcasts using a general search engine. Primary language is query language that echos the words of the person who is speaking in the relevant video segment. We assume that automatic speech recognition (ASR) transcripts will help us match primary language in queries with jump-in points, but that general query language found in ASR-transcripts is less likely to be specifically relevant to the user's information need. We describe each of our algorithms, report results and end with conclusion and outlook.

*At the time the work presented here was done the author was affiliated with Novay, Enschede (The Netherlands) and Delft University of Technology.

MediaEval 2011 Workshop, September 1–2, 2011, Pisa, Italy

2. EXPERIMENTAL FRAMEWORK

In this section, we describe our approaches to RSR. For all runs, we produce our ranked list of jump-in points using a standard IR algorithm to retrieve video fragments that have been defined on the basis of the ASR-transcripts. We return the start point of each fragment as a jump-in point. Fragments are defined as a sequence of sentences of about 40 non-stop-words. Sentences are derived on the basis of punctuation (full-stop = sentence end), which is hypothesized by the recognizer and included in the output of ASR-system. If a sentence is less than 40 words in length, subsequent sentences are added until it approximately meets this target.

Mark Hepple's [2] part-of-speech (POS) tagger is used to tag and lemmatize all words. We remove all closed class words (i.e., prepositions, articles, auxiliaries, particles, etc.). To compensate for POS tagging errors, we additionally remove English and Dutch stop words (standard Lucene search engine stopword lists). Word and sentence segmentation, POS-tagging and term selection are implemented as a UIMA (http://uima.apache.org) analysis pipeline.

We carry out ranking using BM25 [5]. Since fragments may overlap, we calculate *idf* (Eq. 1) on the basis of the sentence, the basic organizational unit of the speech channel,

$$\mathrm{idf}(t) \quad = \quad \log \frac{N - df_t + 0.5}{df_t + 0.5}. \qquad (1)$$

Here, N is the total number of fragments, and df_t is the number of fragments in which term t occurs. The weight of each term in each fragment-document is given by $w(d,t)$,

$$w(d,t) \quad = \quad \mathrm{idf}(t) \frac{(k+1) * f_{dt}}{f_{dt} + k * (1 - b + b * \frac{l_d}{\mathrm{avgdl}})}, \qquad (2)$$

where f_{dt} is the number of occurrences of term t in document d, l_d is the length of d, and $avgdl$ is the average document length. In our experiments, we set $k = 2$ and $b = 0.75$.

The retrieval status value (RSV) of a document for query consisting of more than one word is defined as,

$$w(d,Q) \quad = \quad \sum_{t \in Q} w(d,t). \tag{3}$$

Note that each query word contributes once to the sum, i.e., repetition of query words is ignored.

We create an initial ranking by ordering all fragments by their RSV values (Eq. 3). In order to generate our final results list, we remove all fragments with a starting time within a window of 600 seconds of a higher ranked fragment.

The approaches used by our runs are shown in Table 1. In runs 4 and 5 we use metadata (descriptions, title and

Table 1: Description of RSR runs

Run ID	Query	Fields	Filtering
1	full	ASR	no
2	full + short	ASR	no
4	full + short	ASR + metadata	no
5a	full + short	ASR + metadata	$d_x(q) > 200$
5b	full + short	ASR + metadata	weighted

tags) along with the ASR-transcripts. These runs make use of the BM25 extension known as, BM25F(ields) [6],

$$w(d,Q) \quad = \quad \sum_{t \in Q, f \in F} w_f w(d_f, t). \tag{4}$$

Here, F is a set of fields, d_f the part of document d labeled as field f, and where w_f is the weight for field f. In our experiments we use $w_f = 1$ for the ASR and $w_f = 0.5$ for all other fields. Tests on the development set showed that results are not particularly sensitive to the exact value and we used 0.5 since it gave the best results.

In runs 5, we applied a query word filter built using a corpus of 3,400 requests for video made by users on Yahoo! Answers, cf. [3]. In run 5a, we removed the most frequent words occurring in the corpus from the queries (83 terms with frequency over 200 were removed). In run 5b, terms frequent across requests in the corpus were given lower weights. We implemented this downweighting by replacing Eq. 1 by,

$$\begin{aligned} \mathrm{idf}'(t) \quad = \quad & \alpha \log \frac{N - df_t + 0.5}{df_t + 0.5} \\ & + (1-\alpha) \log \frac{N_{req} - reqf_t + 0.5}{reqf_t + 0.5}, \end{aligned} \tag{5}$$

where N_{req} is the number of requests in the corpus and $reqf_t$ is the number of requests in which term t occurs. In the reported runs we have set $\alpha = 0.5$.

3. RESULTS AND CONCLUSION

Our results our reported in Table 2 and given in terms of the mean Generalized Reciprocal Rank (mGRR) [4] with tolerance windows of 10, 30 and 60 seconds. In general, larger tolerance windows correspond to larger scores. However, whether adding the short query improves performance (cf. run 1 vs. 2) varies depending on the tolerance window used. Note that the statistical significance of this difference remains to be checked.

We can see that filtering or downweighting general query-language words (e.g., *video* and *tv*) can indeed improve results. Downweighting has a larger impact, suggesting that

Table 2: Results reported in terms of mGRR

Run ID	10	30	60
1	0.24	0.33	0.38
2	0.22	0.34	0.40
4	0.23	0.34	0.39
5a	0.24	0.36	0.41
5b	0.28	0.39	0.45

general query-language words should not be treated by extending a conventional stop word list for application in video retrieval. No appreciable difference was observed between using ASR transcripts alone and using both ASR transcripts and metadata in the conventional case in which query words are all treated the same (cf. run 2 vs. run 4). Apparently, separate treatment for different types of query words is particularly important to fully exploit the contribution of metadata (cf. run 2 vs. run 5b). In the experiments, we find that adding the query-language downweighting slightly improves the results of very many queries, as long as they already performed reasonably well without downweighting. However, a number of queries fail completely. An investigation of these cases carried out by hand revealed that failure was in most cases due to vocabulary mismatch between query and target item, suggesting that performance would benefit from the use of conventional techniques for query expansion.

Future work will focus on developing more sophisticated models for general-language query words. Additionally, we will attempt to model of query words that are 'primary', i.e., more likely to occur in spontaneously produced and/or direct speech and less likely to occur in the descriptive or indirect descriptions of the video in the metadata.

Acknowledgments The research leading to these results has received funding from the European Commission's 7th Framework Programme (FP7) under grant agreement no. 216444 (EU PetaMedia Network of Excellence).

4. REFERENCES

[1] J. Besser, M. Larson, and K. Hofmann. Podcast search: User goals and retrieval technologies. *Online Information Review*, 34:3, 2010.

[2] M. Hepple. Independence and commitment: Assumptions for rapid training and execution of rule-based POS taggers. In *ACL*, 2000.

[3] C. Kofler, M. Larson, and A. Hanjalic. To seek, perchance to fail: expressions of user needs in internet video search. In *Proceedings of the 33rd European conference on Advances in information retrieval*, ECIR'11, pages 611–616, Berlin, Heidelberg, 2011. Springer-Verlag.

[4] M. Larson, M. Eskevich, R. Ordelman, C. Kofler, S. Schmiedeke, and G. Jones. Overview of MediaEval 2011 Rich Speech Retrieval Task and Genre Tagging Task. In *MediaEval 2011 Workshop*, Pisa, Italy, September 1-2 2011.

[5] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 1 edition, July 2008.

[6] S. E. Robertson, H. Zaragoza, and M. J. Taylor. Simple BM25 extension to multiple weighted fields. In D. A. Grossman, L. Gravano, C. Zhai, O. Herzog, and D. A. Evans, editors, *CIKM*, pages 42–49. ACM, 2004.