Search Technologies Report

Title: Personal Archive Search
Module code: CA4009
Module name: Search Technologies

Name: *Liam McAweeney*
Student Number*: 14415152*
Email : *liam.mcaweeney2@mail.dcu.ie*

Name: *Declan Lunney*
Student Number*: 15396571*
Email : *Declan.lunney2@mail.dcu.ie*

Date of Submission: 14/12/18

Table of contents

Disclaimer

A report submitted to Dublin City University, School of Computing for module CA4009:Search Technologies, 2018/2019.

I understand that the University regards breaches of academic integrity and plagiarism as grave and serious.I have read and understood the DCU Academic Integrity and Plagiarism Policy. I accept the penalties that may be imposed should I engage in practice or practices that breach this policy. I have identified and included the source of all facts, ideas, opinions, viewpoints of others in the as-assignment references. Direct quotations, paraphrasing, discussion of ideas from books, journal articles,internet sources, module text, or any other source whatsoever are acknowledged and the sources cited are identified in the assignment references. I declare that this material, which I now submit for assessment, is entirely my own work and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

By signing this form or by submitting this material online I confirm that this assignment, or any part of it, has not been previously submitted by me or any other person for assessment on this or any other course of study. By signing this form or by submitting material for assessment online I confirm that I have read and understood DCU Academic Integrity and Plagiarism Policy (available at: http://www.dcu.ie/registry/examinations/index.shtml)

Name(s): Declan Lunney & Liam McAweeney
Date: 04/12/18

Abstract

This project focuses on the development of a new system for personal archive search where users will be able to search for files based on their content. The report will outlay how the new system will solve the problem of not being able to search files on their content. The report will provide a overview of how the system will work from preprocessing, indexing, implementation and evaluation. Also outlined in this report are the benefits of this system. Using new techniques, accurate evaluation of the systems can be performed on user's own archives.

1 Introduction

The system's objective is to improve the search ability of personal archives. The motivation for this system is that current implementations of information retrieval on personal archives are not using the latest technologies.
Before there was a lack of methods for evaluating IR within personal archives which reduced the implementation of new IR techniques. Since further research has been carried out, the implementation of new IR techniques using the latest technologies is more feasible.

Constraint: For evaluations we do not have enough participants to evaluate.

2 Functional Description

The system will use automatic speech recognition to generate textual surrogates to represent the context of the audio files. It will use automatic generation of contextual metadata and optical character recognition to represent videos and image files. This representation of multimedia files will be indexed the same as text files.
The system will use the Okapi BM25+ algorithm to rank documents in terms of probability of relevance.

Assumptions of our system are:

- The textual surrogates and contextual metadata will accurately represent the multimedia files.
- Number of relevance documents that are to be returned from a search query is small.

Analysis of the individual components is included in their respective sections

3 Implementation

3.1 Multimedia

3.1.1 ASR: Automatic Speech Recognition

The system will generate textual surrogates from audio files for the purpose of indexing and searching as if it was a tradition text file. The system will use google cloud's speech video version ASR engine technology due to it being the best ASR speech to text technology in today's market at translating text to audio. [1,2]

3.1.2 OCR:Optical character recognition

The System will be able to extract text from video and images into metadata for the purpose of indexing and search for like a tradition text file. Extracted text will be compared with text from neighbouring frames, taking the most common character at each position in a word[3]. A threshold of 80% dominance in each position of a retrieved word is necessary. If this is not achieved external knowledge graphs will be implemented to find the terms which are most likely to be terms from the frame.

3.1.3 Video Analysis

The system will sample from a video file through implementing shot boundary. The system will implement various techniques to prevent dropped shot boundaries and false shot boundaries. [4] The system will implement automatic generation of contextual metadata and will implement the same processing techniques as MediAssist. [5]. The user can manually enter a speaker's name into the metadata of a video or audio file. Voice recognition will then automatically recognise the speaker in future videos or audio files. [6]

3.2 Preprocessing

The system will carry out preprocessing to identify the optimal form of a term for indexing. This allows the system to more accurately retrieve information based on the user's query.
The following stages will occur in the preprocessing part of our system.

3.2.1 Tokenization

The system will separate all words, numbers and other characters by whitespace, they are known as tokens. The system will adopt Rapid Automatic Keyword Extraction algorithm so the system will be able to extract the crucial term in a given sentence or text. [7] A reason we do this is so the system will eventually evaluate the frequency value of the tokens and remove all infrequent tokens too.

3.2.2 String Similarity

The system will use a string similarity algorithm based of the Levenshtein distance. This will allow the system to be able to work with misspellings or grammar errors that occur in the tokens. [8,9]

3.2.3 Token Normalization

The system will use normalisation where it will be 'canonicalizing tokens so that matches occur despite superficial differences in the character sequences of the tokens'. An archive could contain words that the have the same meaning, eg USA and U.S.A. The system will use equivalence classes [10]. USA and U.S.A. now would be both mapped onto USA.

3.2.4 Stopwords

The system will use a predefined list of stopwords. A stopword can be a word with meaning in a specific language or a token that has no linguistic meaning. It will reduce size of the index file, reduce the memory requirement needed and it will improve search efficiency as there will be no matching done in a user query for stop words.

3.2.5 Case Folding

The system will carry out case folding but only lowercase the words at the beginning of the sentence leaving the words in the middle capitalized. The benefit is the system will be able to match the start of a sentence to match a users query word which could be in lowercase. The standard technique is to lowercase all words but this can come at a cost of losing information [11]. The system will take consideration of how Orange can be a french mobile

company or a fruit and how it would not recognise the company from fruit when lowercasing everywhere.

3.2.6 Stemming

The system will use the second version of the M.F Porter algorithm known as the ‘Snowball’ stemmer as it best suits our system. This stemming framework allows the development of unique stemmers for character sets [12]. The benefit of doing stemming is that it will remove various suffixes, minimise storage and will allow exact matching stems so it allow the system to be able to match a user query with different tenses of words [7,12].

3.3 Systems Preprocessing CFD

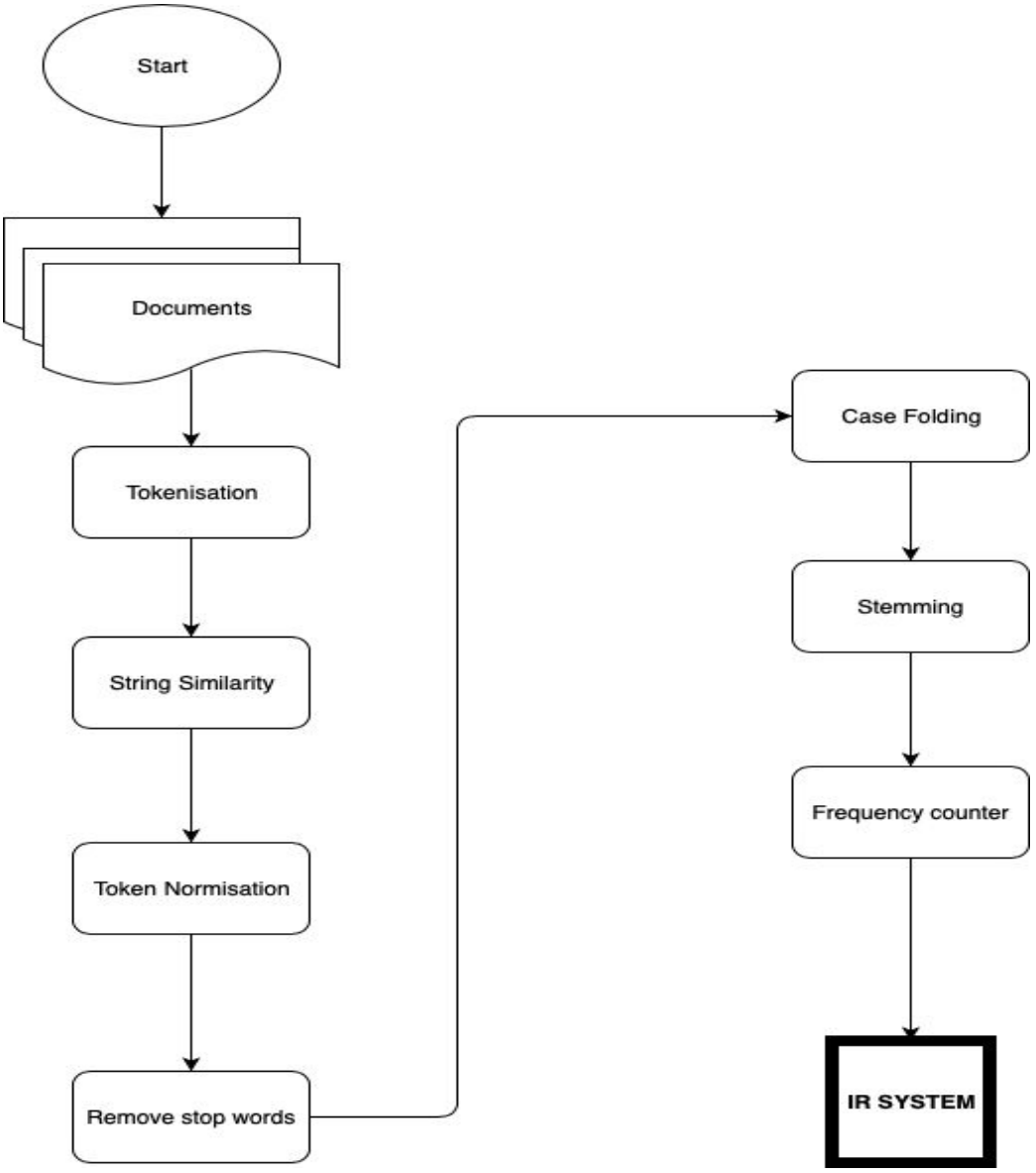

Fig 3.3.1

3.4 Indexing

After preprocessing the system will generate an inverted index of the preprocessed file. Inverted indexing will map terms to documents. Inverted indexing has many advantages[13]. Forward indexing maps documents to terms. Forward indexing is fast as index but slower at searching, while inverted indexing is slower at indexing but quicker at search[14]. Slower indexing would not affect the system as typically personal archives are not very big. This will result in quicker search. Indexing will be perform at a predefined time such as when the system is sleeping, this will reduce the impact of a slower indexer.

Each file type will have their own inverted invert to improve search time for user.

When searching for a given term, first, the system will look up the term in the dictionary to retrieve a pointer to a corresponding posting list in the postings [13]. Then it traverses the posting list and retrieves all documents in the posting list. The system will sort them by term frequency.

The system will adopt Zipf's Law where it will be able to rank the frequency of words [15]. The system will be able to count the frequency of each token.

To increase performance, compression techniques will be implemented by the system. The pros of the system of implementing this is it will use less disk space. Compression could reduce the space occupied by an inverted index by 75%. Faster processing of queries will occur because more of the frequently used parts can be stored in memory[13,16]

3.5 Information retrieval

The system will use a probabilistic information model for information retrieval. This model was chosen over others models as it potential can retrieve the perfect ranked list of documents. The vector space model can result in false positive matches if the search terms do not precisely match document terms.
The system will return a list of documents from the archive sorted by the probability of relevance to the user's query. The system will use the Okapi BM25+ algorithm [17]. This algorithm is an extension of the Okapi BM25 ranking algorithm. "They use the TREC GOV2, WT100g, WT2G, and Robust04 collections and show that BM25L outperforms BM25 in all cases."[18].

$$cw(i,j) = cfw(i) \times \left(\frac{tf(i,j) \times (k_1 + 1)}{k_1 \times ((1-b) + (b \times ndl(j))) + tf(i,j)} \times \delta \right)$$

Fig 3.4.1

The algorithm in figure 3.4.1 is the Okapi BM25 with a constant, denoted as delta, appended to it.[19] “They suggest that a δ value of 1 is effective across collections.”[18]
Personal information archives can contain a large volume of image files. The textual surrogates produced from image files will be shorter than other documents in the archive. This could make the average document length dramatically shorter than text files, such as a news article. This could cause a problem in the retrieval of a document such as a news article as the normalisation of document lengths may cause the top ranked retrieved documents to be images. This is why the system is going to employ the BM25+[17]. “Lv & Zhai observe that the document length normalization of BM25 (Ld/Lavg) unfairly prefers shorter documents to longer ones. ”[18]. The values of b, k and δ were determined through experimentation. This process will be further explained in the next section.

The system will implement relevance feedback in the form of explicit and implicit.[20] The user will perform explicit feedback through specifying if a document was found not to relevant to the query. The user will perform implicit feedback through the length of time spent in a document, eg, if a user spends 4 seconds reading a document. It probably is not relevant.

Users can specify a file type before the run their query if they know what type of document they want. The system will implement a different version of the search algorithm based on the file type selection. This is to reduce search time for users.

4 Evaluation

The evaluation of a personal information archive IR system is difficult due to the development of a test collection. It is problematic for reasons relating to the personal and private nature of the data and associated information needs and measuring system response effectiveness.

To overcome the difficulties outline, the evaluation of this system is similar to the concept of the living laboratory described in [21]. Each participant of the evaluation will perform the evaluation on their own archive. This will ensure that each participant will be evaluating the IR of a personal archive.

A virtual environment will be created and shared with all the participants. This environment will contain the indexing method which will be applied to each of the participants archives. It will also contain a tool which will allow each participant to assess the relevance of items retrieved. The tool will then generate the IR evaluation metrics based on the participants inputs and aggregate the results. This will protect the privacy of each of the participants personal data.

Search topics in IR evaluation are generally pre-defined known topics to be covered in the test collection. Since each test collection will vary between participants it is not known what topics will be covered in each collection, therefore broader search topics must be defined eg meeting a friend. The participant must then define their own specific search topic based on the broader definition.

When performing IR on a personal archive the number of relevant retrieved document will be small. Therefore the mean reciprocal rank (MRR) will used as the evaluation metric.

The participants will perform searches with multiple versions of the BM25+ algorithm with different values for k1, b and δ. All the evaluation metrics will be aggregated for each version of the IR algorithm. The version with the highest average MRR will be used for the IR of the personal information archive.

5 Conclusion

In this work, an overview has been given of preprocessing, indexing , implementation and evaluation of the system. In the case of preprocessing, words are manipulated to optimise their search ability. In the case of indexing, inverted indexing is implemented to optimise search time. In the case of implementation, the probabilistic model BM25+ is used as it caters better for longer documents. In the case of evaluation, a virtual environment will be created to synchronize evaluation the system on user's own archives, on their own systems. The evaluation metrics are then aggregated and analysed. This will ensure the user's data privacy is preserved. Further work would include incorporating a voice search to allow the user to search through speech.

6 References

[1]
Rosales-Huamaní, J. et al., 2018. A Prototype of Speech Interface Based on the Google Cloud Platform to Access a Semantic Website. Symmetry, 10(7), p.268. Available at: http://dx.doi.org/10.3390/sym10070268.
[Accessed 1 Dec. 2018].

[2]
Kincaid, J. (2018). Which Automatic Transcription Service is the Most Accurate? — 2018. [online] Medium. Available at: https://medium.com/descript/which-automatic-transcription-service-is-the-most-accurate-2018-2e859b23ed19
[Accessed 28 Nov. 2018].

[3]
Jie, D., Guotao, Z. and Fang, X. (2018). Research on Video Text Recognition Technology Based on OCR. Xiaogan China: College of Technology, Hubei Engineering University, p.1 Available at: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8337428
[Accessed 28 Nov. 2018].

[4]
Bhattacharyya, Siddhartha & Bhaumik, Hrishikesh & De, Sourav & Klepac, Goran. (2016). Intelligent Analysis of Multimedia Information. 10.4018/978-1-5225-0498-6. Available at: https://www.researchgate.net/publication/298258010_Intelligent_Analysis_of_Multimedia_Information
[Accessed 20 Nov. 2018].

[5]
O' Hare, N., Lee, H., Cooray, S., Gurrin, C., J.F. Jones, G., Malobabic, J., E. O'Connor, N., F. Smeaton, A. and Uscilowski, B. (2006). [online] Doras.dcu.ie. Available at: http://doras.dcu.ie/16138/1/MediAssist_Using_Content-Based_Analysis_and_Context_to_Manage_Personal_Photo_Collections.pdf
[Accessed 15 Nov. 2018].

[6]
Govivace.com. (2018). Speaker Identification | Govivace. [online] Available at: https://www.govivace.com/products/speaker-identification/
[Accessed 14 Dec. 2018].

[7]
Mine your text and extract valuable data! Natural language processing tools. (2018). [Blog] *krakensystem*. Available at: https://krakensystems.co/blog/2018/nlp-syntax-processing
[Accessed 5 Dec. 2018].

[8]
Medium. (2018). String Similarity Algorithms Compared – Appaloosa Store – Medium. [online] Available at: https://medium.com/@appaloosastore/string-similarity-algorithms-compared-3f7b4d12f0ff [Accessed 13 Dec. 2018].

[9]
Zhang, S., Guangrong, B. and Hu, Y. (2017). Research on string similarity algorithm based on Levenshtein Distance - IEEE Conference Publication. [online] Ieeexplore.ieee.org. Available at: https://ieeexplore.ieee.org/document/8054419 [Accessed 13 Dec. 2018].

[10]
MANNING, Christopher D, Prabhakar RAGHAVAN, and Hinrich SCHUTZE. Introduction to information retrieval. 1st pub. Cambridge: Cambridge University Press, 2008, xxi, 482 s. ISBN 9780521865715 available at:https://www.math.unipd.it/~aiolli/corsi/0910/IR/irbookprint.pdf [Accessed 18 Nov. 2018].

[11]
The term vocabulary and postings lists. (2009). [ebook] Cambridge: Cambridge University Press, p.1. Available at: https://nlp.stanford.edu/IR-book/pdf/02voc.pdf [Accessed 13 Dec. 2018].

[12]
Anjali Ganesh, J. (2011). A Comparative Study of Stemming Algorithms. 2nd ed. [pdf] Gujarat, p.1. Available at: https://pdfs.semanticscholar.org/1c0c/0fa35d4ff8a2f925eb955e48d655494bd167.pdf [Accessed 5 Dec. 2018].

[13]
Hadraba, A. (2015). Inverted index implementation. [ebook] p.3. Available at: https://is.muni.cz/th/hsr4u/thesis.pdf [Accessed 13 Dec. 2018].

[14]
Jain, S. (n.d.). Difference between Inverted Index and Forward Index - GeeksforGeeks. [online] GeeksforGeeks. Available at: https://www.geeksforgeeks.org/difference-inverted-index-forward-index/ [Accessed 26 Nov. 2018].

[15]
Chen, Ye-Sho & F. Leimkuhler, Ferdinand. (1987). Analysis of Zipf's law: An index approach. Inf. Process. Manage.. 23. 171-182. 10.1016/0306-4573(87)90002-1. Available at:https://www.sciencedirect.com/science/article/abs/pii/0306457387900021 [Accessed 28 Oct. 2018].

[16]

ZHANG, Jiangong, Xiaohui LONG, and Torsten SUEL. Performance of compressed inverted list caching in search engines. 2007. Available at:http://wwwconference.org/www2008/papers/pdf/p387-zhangA.pdf

[Accessed 28 Oct. 2018].

[17]
Lv, Y. and Zhai, C. (2018). When documents are very long, BM25 fails!. [online] Semanticscholar.org. Available at: https://www.semanticscholar.org/paper/When-documents-are-very-long%2C-BM25-fails!-Lv-Zhai/ad51e702bdcb6957a3b959e7ff97769b52999fc9?fbclid=IwAR2jApWRckZJDL7q-DXQhlVAaZK5hIWgLn_GkpvXxJN8yQdDgbnYa0jxXGE
[Accessed 4 Dec. 2018].

[18]
Zaragoza, H. (2009). [online] Staff.city.ac.uk. Available at: http://www.staff.city.ac.uk/~sb317/papers/foundations_bm25_review.pdf
[Accessed 5 Dec. 2018].

[19]
Trotman, A., Puurula, A. and Burgess, B. (2014). 3. [online] Cs.otago.ac.nz. Available at: http://www.cs.otago.ac.nz/homepages/andrew/papers/2014-2.pdf
[Accessed 27 Nov. 2018].

[20]
Andreu-Mar´ın, A. (2017). [online] Ceur-ws.org. Available at: http://ceur-ws.org/Vol-2125/paper_73.pdf
[Accessed 10 Dec. 2018].

[21]
J. F. Jones, G. and Chen, Y. (2018). [online] Doras.dcu.ie. Available at: http://doras.dcu.ie/16435/1/A_Strategy_for_Evaluating_Search_of_Real_Personal_Information_Archives.pdf
[Accessed 14 Dec. 2018].