CA4009
Search Technologies
Research and Development Project

In Video Search & Summarisation

Computer Applications and Software Engineering IV (CASE4)

Erika Rellermo	1833706	erika.rellermo2@mail.dcu.ie
Joanna Talvo	18342523	joanna.talvo2@mail.dcu.ie
Chloe Ward	18302716	chloe.ward35@mail.dcu.ie
Stefan Lupu	18457016	stefan.lupu2@mail.dcu.ie
Marius-Constantin Senchea	18719785	marius.senchea2@mail.dcu.ie

Date of submission: 23/11/2021

1.0 Disclaimer

A report submitted to Dublin City University, School of Computing for module CA4009: Search Technologies, 2021/2022. 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 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): Erika Rellermo, Joanna Talvo, Chloe Ward, Stefan Lupu, Marius-Constantin Senchea

Date: 23/11/2021

2.0 Table of Contents

3.0 Abstract

The purpose of this document is to show our analysis and implementation of our proposed search system.

In our functional specification, we explain the features that will be implemented. We also go into detail about the various methods and algorithms that will be used to ensure that our extension provides accurate transcriptions and summarizations. This chrome extension is to allow users to essentially “ctrl+f” a video using keywords. This will be done by transcribing the audio of the video into text using Automatic Speech Recognition (ASR) and the text which occurs on screen of the video using Optical Character Recognition (OCR) which is a method of extracting text from video images/frames, then annotating the results with their corresponding timestamps. This will then allow the user to search the transcribed text and then jump into the video where they occur.

We will also be providing video summarizations which will give the users video summarised text which will be implemented with a query based summarization algorithm which will score and group sentences based on their similarity to the query.

In the final section of this report, we go into detail about how the proposed system should behave once successfully implemented. We discuss the various evaluation methods and the user testing that we will conduct to get an understanding of how users will interact with the core functionalities of the search system.

4.0 Introduction

There is an incredibly large amount of data in the form of video being generated online each and every single day. As such systems that can help users search, understand and keep up with this stream of information are becoming increasingly more important. Additionally Covid-19 has put to attention the specific need for such capabilities with regards to online learning. In-video searching and video summarization are particularly useful for students analysing recorded lectures.

The application will be a chrome extension that allows users to search the content of videos in a similar fashion as to how the common keyboard shortcut “ctrl+f” operates, where instead of jumping into the text file, it instead jumps to the timestamp of where the query made by the user is in the video, either spoken or written on screen. Additionally, the extension will give a summary, selecting the main points of the video and presenting them back to the user in bite sized chunks.

Our aim is to implement these systems using OCR and ASR as our foundational algorithms that will transcribe and annotate the video and audio inputs respectively. In the case of video

summarization further processing will be needed. Using an evolutionary query based approach we can select what is the most valuable and relevant content from the data.

To maximise our user experience we need to take into account the various computational constraints present in the algorithms we will use. The queries should be answered quickly and the generation of the summary should happen as fast as possible, preferably instantaneous. Additionally the input transcription, annotations and the generations from the evolutionary query algorithm should take up as little system memory as possible. This would ensure the application can run smoothly and processes can fully execute.

5.0 User Analysis

5.1 Users

In the current situation we are in, people have been adapting to working and studying remotely. Students are given recorded lectures, businesses are having online meetings, organisations are offering more online courses and conferences, and the public are more into video contents than ever before.

During our planning, our key goal was to make sure to devise a system that is accessible to a wide range of users as possible. The target audience of our proposed system is categorized in three types of users: students, businesses and organizations, and the general public.

Since our application will be a chrome extension, this will not require any specialised knowledge from our target audience. The user will simply have to add the extension in chrome in order to use it.

Students

Students can use this application to search for a specific topic they wish to check in their recorded lectures. As students ourselves, we find this to be very effective as most recorded lectures are time consuming to watch. It is easier for us to search for the topic we want without having to manually find it in the recorded lectures. A summarised text of the recorded lectures will also come handy when it comes to revision for upcoming exams.

Businesses and Organizations

Businesses and organisations can also benefit from this application. Online meetings and conferences are being recorded for future use. For example, a newly employed software engineer joins the Search Team and has to learn all the necessary information the team works on. They can watch the recorded meetings or other related video contents that have been talked about and done before. Having the ability to search for a certain topic in a lengthy informative video can save a lot of time, especially in a fast paced work environment. A summarised text

content on business meetings can also be beneficial when it comes to only saving the pivotal parts of the meeting.

General Public

The general public can also use this application to simply search for a topic in a video content they are interested in. Some video contents tend to be very long and some users like to rewatch these videos however not in full. Instead, they tend to skip to the section where they believe is the most interesting part of the video. Our application will save them from skipping since it will take them where their searched topic is found. A summarised text content of a video can be also of interest to this type of user.

5.2 Using the application

5.2.1 Search Queries

To execute a valid search, the user will need to have a search query and a video.

Search query: The search query is simply the object that the user is expecting to find in a video. For example, if a user is currently watching a recorded lecture on compilers, they can search for the topic “semantic analysis” in order to find all the sections in the video where semantic analysis is mentioned or seen on screen.

Video: The user must be able to have video content in the browser. The application will detect that video content is present in order to perform a search query and summarisation.

5.2.2 Search Results

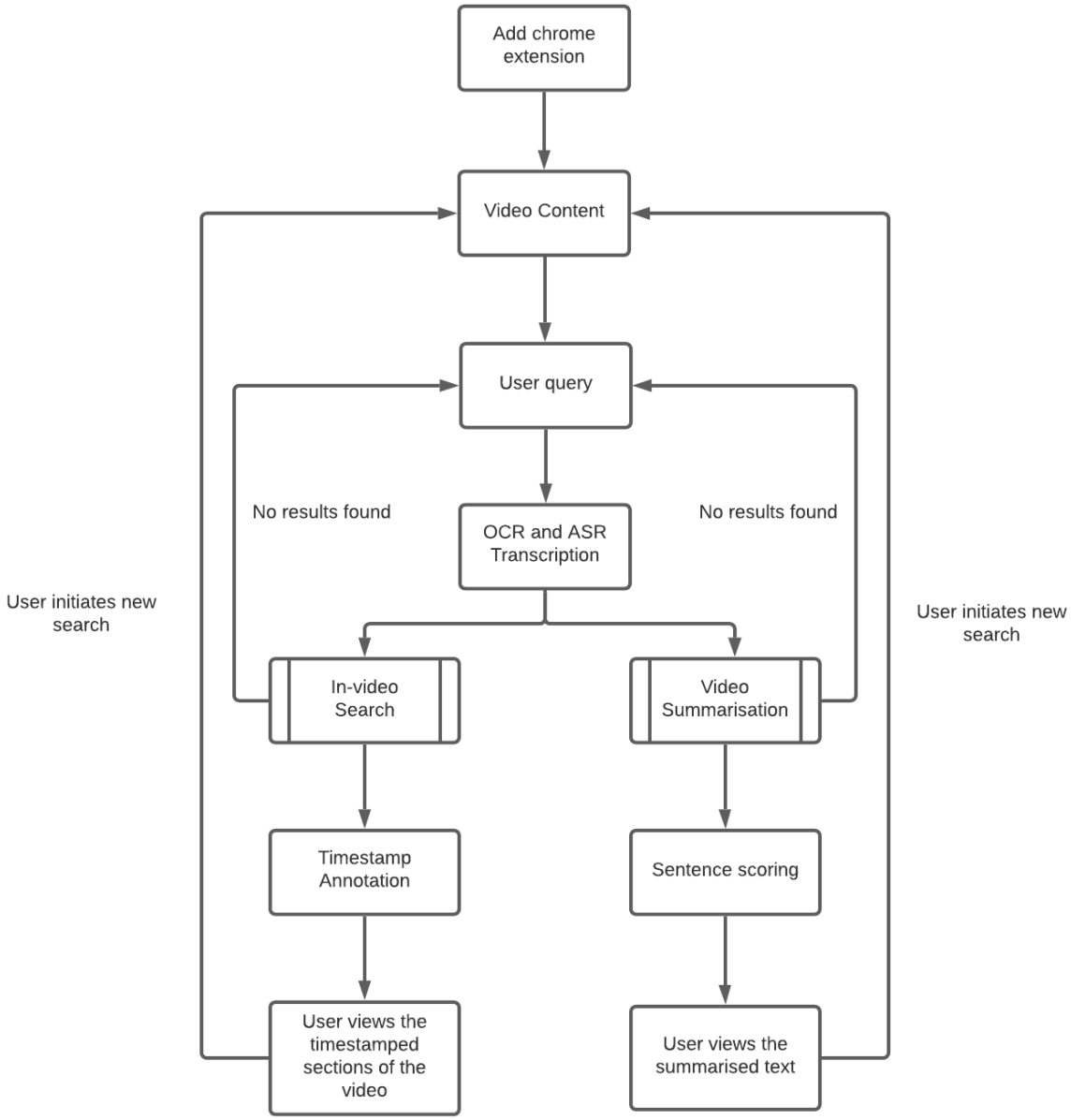
Once a search has been successfully performed, the user will then be given the section in the video where their searched query appears. In the event where the search query has multiple occurrences in the video, the user will be given indications in the video where these occurrences can be found.

In the event that the user wants to get the summarised text content of the video, there will be a pop up of the summarised text and an option to download it.

5.3 Operational Scenarios

Since our application will be a chrome extension, it is ultimately designed to be simple and easy to operate. The sample operational scenario will start with a user adding the chrome extension and from there, the user will perform a search in the video presented on the browser. The user will also get the summarised text version of the video in a pop up manner and an option to download it.

Below is the search operation flow diagram to help visualize the scenario.



Flow of search operation

5.4 Pre-existing Services

Microsoft Stream, Kaltura and Azure Media Analytics are all examples of such service providers.

5.4.1 Microsoft Stream

Using Deep Learning Microsoft provides the ability to search their content in Microsoft Stream.

When searching for videos, Stream finds videos based on what is said in it.

5.4.2 Kaltura

They provide an enterprise video management system which performs a variety of microservices, one of which being in video searching.

5.4.3 Azure Media Analytics

They are a dedicated suite of speech and computer vision services delivered on top of Azure Media Services with functionalities such Video Summarization, motion detection, OCR, text to speech, etc.

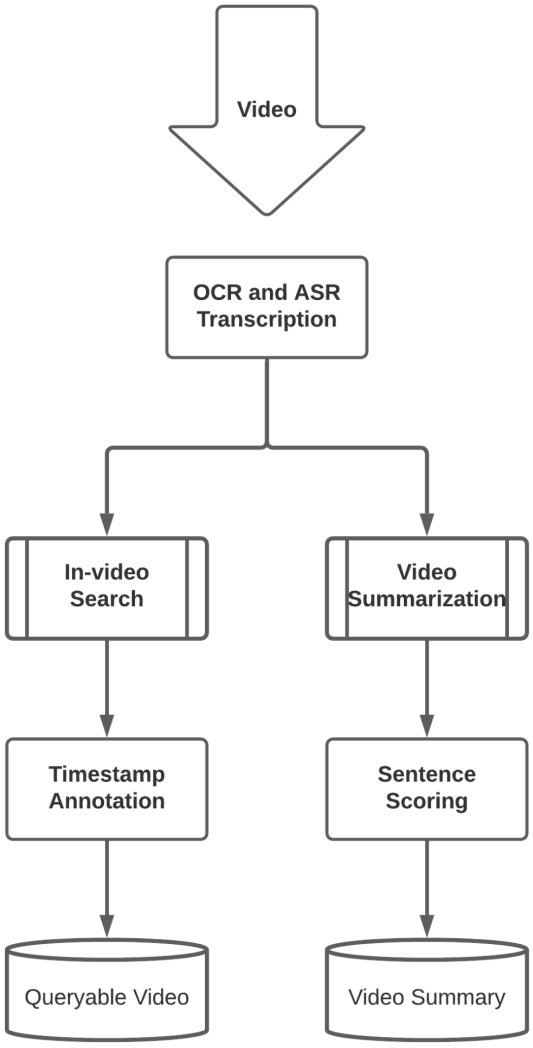
6.0 Scientific Functional Description

6.1 Algorithms

There are a variety of different services which provide in-video search and video summarization capabilities. Both in-video search and video summarization are first and foremost dependent on transcription algorithms to collect and transform the data into a medium on which they can perform further analysis. Usually this medium is text.

In the case of in-video searching the transcribed text is then annotated with the timestamps where the respective words occur. These timestamps are then used to jump into the video chronologically depending on the searched query.

In the case of video summarization, additional algorithms are used to extract the most relevant information and generate a summary of the transcribed text.



6.1.1 Optical Character Recognition

Optical Character Recognition (OCR) is the conversion of images of typed, handwritten or printed text to transcribed text. On a high level, our program will feed in video frames to an OCR algorithm in order to extract text that may be present.

There are 3 main stages in OCR:

1. Pre-Processing
2. Character Recognition
3. Post Processing

6.1.1.1 OCR Pre-processing

Preprocessing consists of many techniques [1]. The most basic 5 of these techniques which we will focus on are:

1. Skew Correction
2. Binarization
3. Noise Removal
4. Thinning and Skeletonization
5. Segmentation

Skew Correction

Skew correction deals with the alignment of text. In many cases, text is not perfectly in line (for example if a presenter would write on a white board, it may go out of line). This stage of preprocessing corrects for these skews. For our application, we shall use the projection profile method as this is the most widely used and simplest of these skew correction methods. The projection profile algorithm will apply different rotations to the video frame and select the frame which scores the best based on a hit based scoring method.

Binarization

Binarization is the process of converting a colored image to black and white. Our program will use adaptive binarization, which works by looking at features of neighboring pixels and applying thresholding in order to convert to a clear black and white image.

Noise Removal

Noise removal is important as there may be some outlier pixels, and light and dark spots on the frame image. These can cause problems for the character recognition step of OCR. The method we will use for noise removal is bilateral filtering, as it is quicker than other methods, and is highly effective in preserving character edges.

Thinning and Skeletonization

The size and width of text can be highly variable at times, especially with handwritten text. The purpose of this pre-processing step is to get all of the input texts to be similar in size and thickness in order to get a consistent conversion. We will use the OpenCV library to apply this skeletonization.

Segmentation

Segmentation is carried out on an image in order to single out characters to be processed.

First we will segment the lines of the image using line level segmentation, which will split text into lines. Next we will perform word level segmentation which will single out words. Finally we will carry out character level segmentation, which will single out the characters. These character segments can then be passed into the next step of the OCR process.

6.1.1.2 Character Recognition

Once the image frame has been pre-processed, it can now be used as input to the character recognition step.

We will use feature extraction in order to recognise our text. We will use feature extraction as the on screen content of the video inputs can be of any type (whiteboard presentation, powerpoint slide, etc). Feature extraction works better on natural handwritten characters over other algorithms so using it would be a catch all method.

Feature extraction [2] works based on the fact that different characters have different features. It uses global or statistical features to detect points in characters. Some techniques in feature extraction are: **moments, zoning, projection, histograms, n-tuples, crossings and distances.** These techniques all look at different features of a character and can all be used to map it's features out.

Classification is then performed, which will compare the feature vectors from the above techniques to be matched with machine learning training patterns.

6.1.1.3 Post Processing

We will then perform post processing on the output. We will attempt to find errors and perform corrections based on different aspects of the output. We will implement the simulated annealing post processing test correction system developed by Gitansh Khirbat from the University of Melbourne [3].

6.1.2 Automatic Speech Recognition (ASR)

Automatic Speech Recognition is an entirely machine based process which decodes and transcribes oral speech into text to make it searchable. ASR is a technology that allows computers to interpret human speech.

The process of how ASR works is that a user speaks into a device and an ASR software detects the speech by breaking down the user's audio into individual sounds. It will then analyze the sound by using algorithms to the spoken word to give a suggestion of the word that was likely spoken.

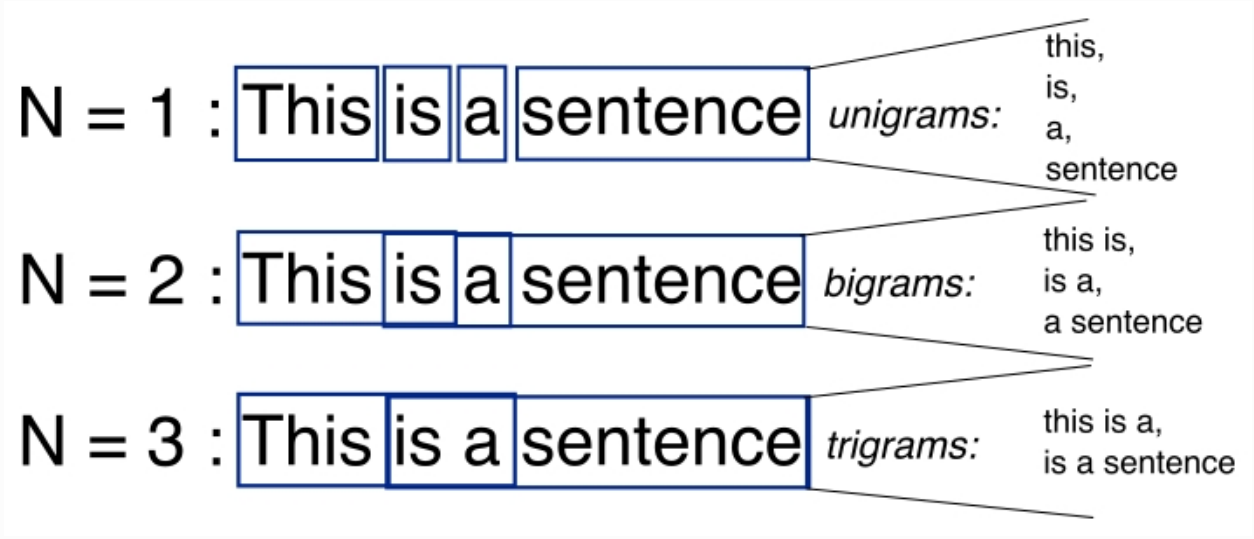
A wave file of the words detected is then created which is then cleaned to eliminate the background noise and normalize the volume[4]. This wave file is then broken down into phonemes which consist of sound blocks such as “th” and “ka” etc [5].

These are the building blocks of sound and language. It then breaks down the cleaned wave file and analyzes it using sequences. The ASR system then analyzes it using a pattern, model or an algorithm and then an output is then produced in the form of text.

There are many challenges with using ASR to transcribe audio into text. A video that a user wants to transcribe to text will have various background noises, as videos are rarely ever recorded in an environment with no noise. This would ultimately reduce the Mean Average Precision (MAP) of the system and allow the system to miss out on some words which were spoken. To combat this problem and increase the MAP of the system, the ASR softwares training data will include background noises such as people talking, electrical noises etc. This would allow the ASR models to isolate the correct sounds to transcribe.

To also further eliminate errors in the transcribed text, we would use a language model which encloses the likely ordering of the words that are generated. An example of this is an n-gram model. [6] Examples of n-grams are bi-grams (“Big House”), tri-grams (“A big House”), and quad-grams (“A big house and garden”). This would then be generated to a pronunciation model in that ordering which would then be translated to audio waveforms. [7] Then when a user uses our chrome extension we will then try to model the following formula to find the most probable sequence of text.

*Pr(audio|transcript)*Pr(transcript)*



N-gram diagram

6.1.3 Stop words

In the case of our system, it will be a chrome extension which will allow a user to transcribe the text of any given video. In a case where a video has been successfully transcribed, we plan to use stop word removal which is available in many libraries. Stop words are the most common words that all search engines avoid to allow them to save space and time. We will apply this to our application by allowing it to avoid these words.

[8] Common examples of stop words are “a”, “the”, “is”, “are” etc. But for the purpose of our chrome extension we will focus primarily on stop words such as “hmm”, “uh”, “um” etc to allow a strong and accurate audio transcription.

6.1.4 Query Based Summarization

During our research, we compared various extractive techniques for text summarisation and Query Based Summarisation (QBS) is what we believe is the best. Among these techniques, they all offer a similar approach but the advantage of QBS is that it keeps redundancy to minimum. To say that a summary system is good, it should be able to extract from diverse topics while trying to minimize redundancy [9]. Redundancy is reduced by grouping similar sentences and the best sentences are selected from each group to produce a summary.

A query based summarization system is used to create a summary providing insight into the contents of a given document. A user will provide a query and the system will return a summarized version of the video’s transcribed text which is related to the user’s given query.

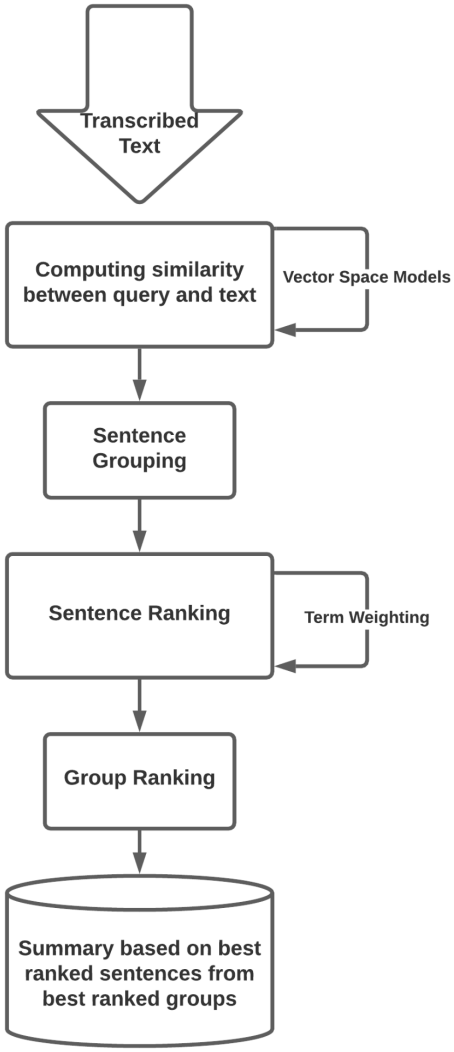
To implement an evolutionary query based summarization algorithm [10] we first begin by calculating the similarity between the sentences in the given document text and the user’s query. This will be done using a vector space model. Based on the similarity scores, similar sentences will be placed into groups.

Within these groups, sentences will be individually ranked on their similarity scores and every word that appears in the document will be scored using term weighting. Term weighting will be done through the computation of term frequency. This concept means that the more often a term occurs in a given document, the more likely it is to be relevant for that document.

The location of each term will also be calculated towards their final scores as terms with close proximity are more likely to be semantically related. Biword indexes will then be created by grouping terms into pairs of two and considering each bi-word as a phrase. An inverted file will then be built to store details of which groups contain which biwords. To optimise the term proximity calculation, stop words will also be removed. This also allows for the efficient breaking down of longer phrases.

Each group’s collective scores will then be combined and computed. The groups will be ranked and listed in ascending order, depending on their group scores. The outputted summary will then be composed from the best scoring sentences from the best scoring groups.

In order to create a more accurate summarization, the outputted summary is used to form a new query. The process is repeated, creating an evolutionary algorithm using the scoring function as a fitness value.



6.1.5 Vector Space Models

A vector space model is a form of best match search that sorts documents based on their similarity to a given search query. Documents are represented and co-ordinated onto a plane where their relevance values are calculated by the following formula:

$$Sim(q, d(j)) = \frac{\sum_{i=0}^{I-1} w_q(i).w_d(i,j)}{\sqrt{\sum_{i=0}^{I-1} w_q(i)^2}\sqrt{\sum_{i=0}^{I-1} w_d(i,j)^2}}$$

Where:

- w is the weight
- q is the given query
- d is the document

Therefore:

- d(j) is the document at index j
- wq(i) is the weight of term i in the query vector
- wd(i, j) is the weight of term i in document vector j

7.0 Evaluation

7.1 Core Evaluation Metrics

Across all of our tests, we will have 3 main performance metrics that we will closely monitor.

Precision: Precision is very important across all of our algorithms, including OCR, ASR and Query Based Summarization.

Recall: Our Query Based Summary shall need to be highly accurate in what parts of the content to recall and summarize.

Accuracy: All aspects of our algorithms will need to be accurate. Without the guarantee of accuracy of our core algorithms, our product will definitely suffer.

7.2 User Testing

In regards to video search, we will mainly test this feature through user testing. User testing is the perfect test for this feature as we can potentially find any bugs we may have missed while developing our system. We would mainly be looking for our testers to try out our extension and report how intuitive and easy it is to use. We will also look for feedback in regards to whether or not our OCR and ASR transcriptions are accurate. We also will investigate whether the words the user will search are reliably found and correct.

For video summarization, we would mainly be looking for users to rate the quality of the summaries that are created. We will also ask the users to try and create a summary of the video themselves and compare their summary with the one our extension will generate. This will help us verify that our summary algorithm will output what our general users would want.

7.3 Acceptance Testing

Acceptance testing would involve scoring runs of our algorithm against a specific percentage of acceptance, say >97% accuracy. This accuracy goal would be measured in multiple stages of our extensions' outputs. The first and most important part of our acceptance tests will involve our ASR and OCR algorithms, and comparing their generated outputs to manually transcribed scripts of the videos we will be testing against. Our algorithms may need to be tweaked in order to achieve our acceptance criteria.

Next, our generated summaries will be tested against human created summaries of the content. These summaries will again be compared to each other and an acceptance rating will be generated. Since these summaries can vary per person, a lower acceptance percentage will be used of around 80%. We will also measure the length and quality of the machine generated summaries and make sure they are of reasonable size and complexity.

8.0 Concluding Section

Our overall product will take the form of a chrome extension for in-video search and video summarization.

Users will provide the video and the system will transcribe the video's audio wave files into accurate text using Audio Cleaning and N-gram and Pronunciation Models to get the most probable sequences of text.

On screen text will then be pre-processed through Skew Correction, Binarization, Noise Removal, Thinning and Skeletonization, and Segmentation. We will then use character recognition using Feature Extraction to recognize text, and Classification to compare feature vectors to be matched with machine learning patterns. The process is then finalized using Simulated Annealing.

As text is transcribed, it will be annotated with its corresponding timestamps. Stop words will also be removed for all processes to account for term irrelevance.

The additional feature of video summarization will then utilise the transcribed text to provide the user with a pop up of a generated summarization based on their given query. This process will use Query Based Summarization, Vector Space Models, Term Weighting, and Inverted Files using Bi-Words.

The target audience for this system are students, businesses, or even general users. It can be used to accurately retrieve information for studying, researching, or even skipping unwanted sections of a video.

To use the in-video search feature, users will provide a query and an indication of the relevant information will appear in the video. To use the video summarization feature, users will provide a query and a pop-up summarization will appear, as well as an option to download it.

To test our ASR, OCR, and summarization features, users will be comparing the system's results with their own manually transcribed texts. They will observe the accuracy of the extension, and examine the relevance of the given summaries.

To finalise the product, we will run acceptance tests which will be given a 3% failure allowance. Similar to the user tests, we will analyse the system precision by comparing the generated texts with our own manually transcribed text. We will also analyse in-video search indication accuracies.

9.0 References

[1] Y. Alginahi, 'Preprocessing Techniques in Character Recognition', 2010. doi: 10.5772/9776.

[2] R. Verma and D. J. Ali, 'Survey of Feature Extraction and Classification Techniques in OCR Systems', *Int. J. Comput. Appl.*, p. 3, 2012.

[3] G. Khirbat, 'OCR Post-Processing Text Correction using Simulated Annealing (OPTeCA)', in *Proceedings of the Australasian Language Technology Association Workshop 2017*, Brisbane, Australia, Dec. 2017, pp. 119–123. Accessed: Nov. 21, 2021. [Online]. Available: https://aclanthology.org/U17-1015

[4] 'How Does ASR Work? The New Generation of Transcription', *Verbit*, Dec. 26, 2019. https://verbit.ai/asr-and-the-next-generation-of-transcription/ (accessed Nov. 21, 2021).

[5] 'Automatic Speech Recognition Technology | by suraj mishra | Alexa Developers SRM | Medium'. https://medium.com/chefs-doeuvre/automatic-speech-recognition-technology-29165e1b7c31 (accessed Nov. 21, 2021).

[6] 'What is Speech Recognition?', Aug. 18, 2021. https://www.ibm.com/cloud/learn/speech-recognition (accessed Nov. 21, 2021).

[7] 'Automatic Speech Recognition (ASR) — Introduction'. https://docs.nvidia.com/deeplearning/nemo/user-guide/docs/en/v1.0.0b4/asr/intro.html (accessed Nov. 21, 2021).

[8] C. Khanna, 'Text pre-processing: Stop words removal using different libraries', *Medium*, Feb. 10, 2021. 23/11/2021https://towardsdatascience.com/text-pre-processing-stop-words-removal-using-different-libraries-f20bac19929a (accessed Nov. 21, 2021).

[9] M. A. R. Deshpande and L. L. M. R. J, 'Modified Text Summarization Based On Information Retrieval', *Int. J. Eng. Res. Technol.*, vol. 1, no. 10, Dec. 2012, Accessed: Nov. 23, 2021. [Online]. Available: https://www.ijert.org/research/modified-text-summarization-based-on-information-retrieval-IJERTV1IS10266.pdf, https://www.ijert.org/modified-text-summarization-based-on-information-retrieval

[10] ‘Evolutionary Algorithms for Extractive Automatic Text Summarization - ScienceDirect’. https://www.sciencedirect.com/science/article/pii/S1877050915006869 (accessed Nov. 23, 2021).