Linking segments of video using text-based methods and a flexible form of segmentation

How to index, query and re-rank data from the TRECVid (Blip.tv) dataset?

Master's thesis

by

Johannes Wassenaar

Under supervision of

Human Media Interaction – University of Twente, Netherlands

Dr. R.J.F. Ordelman

Dr. D. Hiemstra

(Dr. R. Aly)

April, 2018

UNIVERSITY OF TWENTE.

Abstract

In order to let user's explore, and use large archives, video hyperlinking tries to aid the user in linking segments of video to other segments of videos, similar to the way hyperlinks on the web are used – instead of using a regular search tool. Indexing, querying and re-ranking multimodal data, in this case video's, are subjects common in the video hyperlinking community. A video hyperlinking system contains an index of multimodal (video) data, while the currently watched segment is translated into a query, the query generation phase. Finally, the system responds to the user with a ranked list of targets that are about the anchor segment. In this study, the payload of terms in the form of position and offset in Elastic Search are used to obtain time-based information along the speech transcripts to link users directly to spoken text. The queries are generated by a statistic-based method using TF-IDF, a grammar-based part-of-speech tagger or a combination of both. Finally, results are ranked by weighting specific components and cosine similarity. The system is evaluated with the Precision at 5 and MAiSP measures, which are used in the TRECVid benchmark on this topic. The results show that TF-IDF and the cosine similarity work the best for the proposed system.

Summary

Audiovisual media are a large part of current day internet traffic. Video will be 82% of the internet traffic by 2020 according to Cisco. We all use digital video means to connect with people by live videos, share our knowledge in how-to videos on Youtube or re-watch missed television programs using video-on-demand websites. A practical example of media usage and archiving is the large archive of multimedia content at Sound & Vision, the Dutch cultural media heritage. It houses an archive containing 18 petabytes of audiovisual data - and it grows every day as current day media flows in as well as older media is digitalized. With all this video content saved and consumed every day, it is important to research how to make this multimodal complex data accessible for both regular users and professionals.

The importance of research in the area of multimedia information retrieval is supported by the ongoing MediaEval and TRECVid benchmark evaluations. The start of different research tasks at these benchmarks motivates teams to work on well-defined tasks such as the Search and (video) hyperlinking task. Video hyperlinking in this context is an automated form of relating media fragments from a video archive to other media fragments based on the multimodal (speech, visual concepts, metadata) information in the anchor fragment. An anchor is a fragment defined by the video start time and end time for which users might want to view other related content, similar to an anchor keyword in Wikipedia documents. Research shows that users are often confused and unable to foresee what content is available in a large collection, such as the archive at Sound & Vision. Video hyperlinks give an alternative way of navigating, exploring through such large archives next to the currently available, but limiting search tools which only search through metadata. The novelty in video hyperlinks is that links are created at the level of media fragments, using the multimodal nature of the source. The use case therefore is to enable users to explore an archive using fragment level links. Another use case is a storytelling form of navigation, where media fragments of some topic are linked together. The resulting targets for a certain anchor fragment should be "about what is represented in the anchor" - sometimes referred to as "topically related", and not content that is "based upon it, which is similar to it".

Video hyperlinking is defined using four stages: 1) anchor identification, 2) anchor representation, 3) target search and 4) target presentation. For each of these stages there are a multitude of problems to be solved but this study focuses on the second and third step. Apart from the steps above, a search system needs to be available to perform the target search step on. In this search system, all the video's data is saved and indexed (made searchable). The indexing process (the way/format the data is saved) is a difficult process on itself, as the data can be saved using different kind of formats and on different search systems.

One problem that applies to the indexing stage is that of segmentation. Data needs to be in some pre-defined form before saving in the search system's index. Modern search engines view their data as documents. But in video's, what is a document? Is it a whole video, speech segment, video shot etc.? Because there are multiple ways to do segmentation (and other ways could be added in the future) and the chance that relevant content falls just outside a segment or spans over multiple segments, flexibility on the segmentation could be an important aspect in order to improve target results of the system. Another important aspect of the indexing strategy is time-code access to the relevant content: without much hassle, users should be taken to the relevant content time in the video, therefore access to this variable within the results is a requirement.

The next problem is generating a sensible query for the search system. This is the main topic of the anchor representation step in video hyperlinking systems. There is a multitude of data available in an anchor and more specifically: the data is multimodal. The problem on how to interpret, combine and summarize this data into one query is challenging. One specific area that is difficult is the audio stream, transcripts often are long streams of text, from which the most important words need to be selected in order to reduce noise that would occur if just all words would've been used. Two methods are proposed: 1) using statistics: with TF-IDF words get a score that reflects their importance for a document in the collection and 2) using grammar with a Part of Speech tagger. Words are tagged and based on grammatical rules keywords in a sentence can be determined.

Lastly, the study focusses on the third step of the video hyperlinking stages, target search. The problem herein lies with the flexibility imposed to the system at the indexing stage. The system needs to produce useful targets, defined by constraints from TRECVid such as length between 10 and 120 seconds. The system returns at the target step a set of result documents – in this case these documents are representations of full videos that match the query. The system needs to create segments in order to fit the constraints as well as to be

useful for the users. The next problem is re-ranking the created segments, because the video (search engine) ranking doesn't necessarily reflect the fragment ranking. E.g. The search engines' top result might have its relevant content scattered over multiple segments, while the 10th result has all the content summarized in a segment of 100 seconds and is thus, following the constraints, much more relevant. Therefore, some way of re-ranking fragments is needed. In this study two methods proposed, firstly - a simple boosting factor based on properties such as words and position in the segment and secondly - making use of the TF-IDF calculations, by calculating the cosine similarity between the target vector and the query vector.

The following research questions follow from the problems mentioned above:
Q1 How to represent and index the multimodal video data so that there is flexibility in the segmentation and time-code access to video segments is possible?
Q2 What performs better for query generation from speech: using TF-IDF or Part-of-Speech?
Q3 What performs better for re-ranking sub-segments: using selective weighting or cosine similarity?

In order to answer the questions a prototype hyperlinking system was build based on an implementation idea from Robin Aly. From his expertise at the Axis project, he came up with an idea to use the term's position and offset parameters in Elastic Search to add time-based information to each term. The position parameter is not used for the actual position of a term, but for the time position of a term in a transcript of a video. This way, segmentation can be kept flexible. Other implementations based on a strict segmentation were also implemented (using parent-child index or using multiple indexes for full videos and segmentations).

To answer the remaining questions, four runs (versions of the system) have been developed, all runs being variations and ideas on improvement based on the run before. In order to reduce over-fitting and tuning of variables too specifically on the anchors, the system was developed and tested using the provided development anchor set. The final runs were evaluated using the official test set.

The TRECVid Videohyperlinking task is evaluated by pooling the top 5 results for each anchor in all the participants' runs. Crowdsource workers on the platform Amazon Mechanical Turk are asked to assess the relevance of the participant's proposed targets. The precision at 5 and MAiSP measure are calculated from the assessments to compare and express performance of the participants' runs. The precision at 5 measure shows the average relevant results in the top 5, while the MAiSP measures segment precision: the effort that a user needs to put in to find the relevant content, measured in how well the proposed target segments fit over the relevant content.

The implementation of the prototype using the term payloads shows that the problem of being segment-free can be solved with Elastic Search, the index is flexible in the sense that returned results can be processed in any way wanted. But in order to facilitate time-code access, the index has to be set-up with an adaptation of the term-vector functionality using a plug-in. The results of the evaluation of the runs using the TRECVid relevance assessments show that in terms of query performance, basic forms of TF-IDF based methods on the audio transcript give the best scores from the runs, while the precision @ 5 measure stays behind on all runs.

The low result of the precision at 5 measure for all the runs could be explained by the fact that the results should be evaluated in the same manner as the other participants, by assessing the top 5 using crowd sourcing on the Amazon Mechanical Turk platform. It could be that some results in the top 5 are not yet assessed for relevance, and thus lowering the score of the system. It would be interesting to see the results when participating in the benchmark. In order to look into this issue, a small investigation has been taken to check how many results of the runs from this system are new and how many are assessed. Next to that, there could also be a problem with over-tuning the parameters. It turned out after investigation, that many of the results from the runs were not assessed and therefore marked not relevant.

Further work could be improving the results by applying more multimodal solutions. Other participants in the benchmark who applied multimodal solutions have generated better results which could indicate that it might be helpful in the performance. Next to that, a research on finding the user's information need: what would the user want to know after viewing a relevant target? More in-depth information could be used for building the query; such as, history of the object/event visible/mentioned, how the object visible/mentioned is made, etc. Currently, the system just uses the mentioned information as a basic query; while more user-adapted queries could be created that fulfill users' needs more effectively.

Contents

1. Introduction

1.1. Introduction

Today, the internet is overflowed with multimedia content. We are even displaying internet media content on television and in front of us on our mobile phones. We stream our own personal lives live using Facebook Live, Instagram and/or Snapchat. Everyday new content is created, whether professionally or not. The popular website for videos “Youtube” grows fast with 300 hours of video uploaded every minute[1]. With 3,25 billion hours of video watched each month in 2016, the website reaches more users "than any cable network in the US"[2]. Media is on the internet one of the largest forms of data traffic according to Cisco's internet usage trends. They show an expected increase in video traffic from 70 percent in 2015 to 82% in 2020[3]. Multimedia is consumed on the internet for entertainment purposes such as Video on Demand (Netflix), re-watching missed shows (local TV pages), for the purpose of following news and learning purposes such as watching tutorials or lectures. Facebook-users are allowed to upload videos as well: users on the popular social networking site are posting 75% more videos than a year ago (2015)[4] and it is reported that users generate 1 milliard views per day[5]. In addition to regular video-watching, live streaming is also an emerging concept. Businesses as well as regular people use live streaming to interact with their followers up to large-scale traffic. An example are gamers using the popular website “Twitch” to stream themselves playing games, but Facebook and Youtube offer live services as well.

With all this media consumed every day, topics on the accessibility in terms of finding, exploring and interacting with these large quantities of multimedia data are important. On the popular video website Youtube, users can access material using different ways: a) direct access, content they found elsewhere, b) searching, finding specific videos and c) browsing, the user looks for interesting content by browsing channels based on recommendations [1]. For this last case, Youtube also has automatic topic channels that are filled algorithmically with videos around a topic, enabling browsing by topic. The access of multimedia is an active research area. Benchmark evaluations on this topic underline the importance of researching multimedia retrieval. Large and long-running benchmarking conferences in both the EU, MediaEval[6] and US, TRECVid[7], which is sponsored by the National Institute of Standards and Technology, are hosting tasks to benchmark and evaluate work in all sorts of multimedia retrieval scenarios, such as searching and linking videos, which started as a "Brave new task" in MediaEval 2012 [2]. The benchmarks allow researchers to work on specific topics, prototyping ideas and evaluating them together with a well-defined dataset in a laboratory setting [3]. The benchmarks evaluate research in video retrieval, especially automatic content-based approaches, as producing manual annotations on large archives is too time-consuming. The cooperation of TRECVid with the research community and stakeholders gives researches real-world tasks and data to work on.

---

[1] “Statistic Brain” http://www.statisticbrain.com/youtube-statistics/ [Accessed 15 Juli 2017]
[2] “Youtube Press” http://www.youtube.com/yt/about/press/ [Accessed 15 Juli 2017]
[3] “Cisco Visual Networking Index: Forecast and Methodology, 2015-2020” [2017]
[4] “Advertising Age, T. Peterson” http://adage.com/article/digital/facebook-users-posting-75-videos-year/296482/ [Accessed 15 Juli 2017]
[5] “Facebook 3Q 2014 Earnings Call Transcript http://files.shareholder.com/downloads/AMDA-NJ5DZ/3804287865x0x789501/CB3B5986-FD59-4607-BCAA-644F9CD63027/Facebook3Q2014EarningsCallTranscript.pdf [Accessed 15 Juli 2017]
[6] More information about MediaEval at http://www.multimediaeval.org/about/
[7] More information about TRECVid at http://trecvid.nist.gov

A practical example where the problem of access to large multimedia archives plays a role is the Netherlands Institute for Sound & Vision. Sound & Vision is the cultural heritage of Dutch television, the archive contains up to 18 petabytes of audiovisual data since the digitalization of older media [4]. Current day born-digital Dutch programs as well as digitized media are acquired and added to the archive daily. Sound & Vision also has a museum to experience the history and social impact of media. Technical solutions that help visitors and clients explore or find interesting data is important research work which is also on the policy of Sound & Vision: research shows that users often take two and a half times more when ordering a fragment instead of a full program. Although the most frequent searches are program titles, they only account 6%, all the others are unaccounted-for queries [3]. Also, some users don't know what to search for or are unable to oversee the large archive, asking themselves what is available here? This thesis combines the practical problem at Sound & Vision with research in the context of the TRECVid benchmark.

Unfortunately, the fact that users cannot oversee large collections of video and often look for fragments instead of full programs shows that current day searching tools are not always a good match with users and the differing use-cases. For navigation and exploring, requirements can differ against requirements for specific searches. Users either know exactly what they are looking for (known item queries) or only have a vague idea [3]. The advances in video hyperlinking offer an alternative way for navigating an archive, supporting exploration beyond retrieval. The last group of users, those who only have a vague idea are the main target. Offering interesting and serendipitous results that are available in the archives that are relevant to what they are currently watching or searching could be helpful. For the other, first group, offering serendipitous results could give them new insights in the availability of materials they didn't think about before, however, it could distract them. Therefore, the results should still contain best matching results to answer their initial query and thought should go to design an interface that decreases distraction as much as possible. In order to support these uses, video hyperlinking takes a slightly different approach. Instead of seeing videos as a whole, recommendations based on everything that is there, video hyperlinking is operating on the level of media fragments. This enables traversal through data using a link structure at the level of fragments [5]. This work shows the steps in building a video hyperlinking system, which could possibly be applied to the S&V archive in the future. The practical application of this system or the techniques used could be very interesting in a real-world scenario.

Figure 1: Video hyperlinking operates on the level of media fragments Source: TRECVid 2016 Task page

The following sections introduce the problem statement, give a background on the problem, as introduction on chapter 2; and give the research questions. Finally, chapter 2 is closed with definitions and limitations.

1.2. Statement of the Problem

Video hyperlinking is a concept similar to linking text documents on the web such as on Wikipedia. Important terms on Wikipedia (and other websites as well), have a blue type and are underlined to signal the user of a link (see image). This link leads the user to other pages that provide additional content. The blue

Figure 2: A hyperlink. Source: https://artbiz.ca/create-meaningful-hyperlinks/

underlined word is the anchor of the link. The page that is linked to is the target. These links are placed in the text by contributors. Automatic anchoring on text such as for Wikipedia has been a topic of interest for research [6], in order to aid in that, often time consuming, process. Because there is more data available than just text in video, video hyperlinking systems often use video fragments as anchors instead. This is so different from traditional linking for users that it introduces difficulties going beyond the scope of this work - in the presentation of the anchor to the user. Next to that, when thinking about that presentation – there are multiple modalities in the anchor segment: visual objects, speech, faces etc. that could be linkable to other sources. It has been difficult to define exactly what users want from a linking system operating at video segment level [7]. Therefore, an assumption has been made that an anchor has one topic, a topic that the uploader intended to convey using the video contents with its multiple modalities [8].

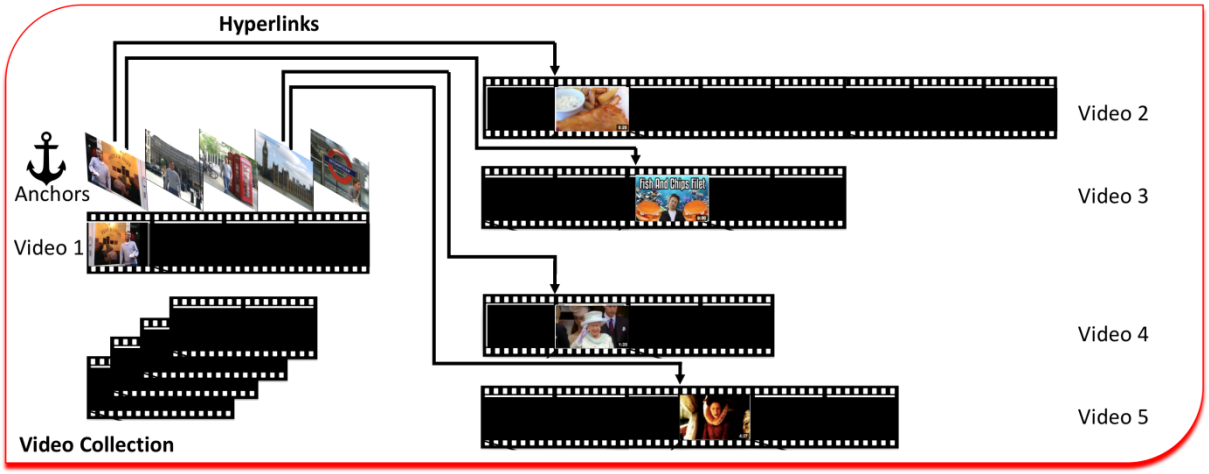


Figure 3: Video Hyperlinking scenario. Source: TRECVid 2017 task page

Currently, when searching or watching videos on the web, websites show recommended videos based on textual features such as the title, description or added tags. One of the problems with this is that these recommended videos could be of such a length that it is inefficient for users; they might need to watch one hour of video to come to the part that involves their information need. Furthermore, specific content inside the videos is not found because there are no annotations, the system doesn't know of the actual content and relies on the user-added descriptions. The aim of a video hyperlinking system is to provide the user with targets, not necessarily being full videos, that are *about* the anchor segment [9], therefore being related to and not similar to the anchor (similarity would not introduce the user to new content).

Figure 4: Current recommendation links. Source: Convenient Discovery of Archived Video Using Audiovisual Hyperlinking (2015)

The main problem statement for a video hyperlinking system is formulated as follows: given an anchor X (video, start time, end time) return a ranked list of relevant targets about X (video, start time, end time, score).

When building a video hyperlinking system, there are a couple of processes needed (the complete flow is depicted in figure 5, while a more elaborate explanation is given in chapter 1.3, Background. This is merely just an introduction). The first thing the system should do is find out what content is in the anchor segment. In order to do so, the system needs to look up that data from somewhere. In the dataset used (see chapter 3 for more details), the anchors contain the following data: video level data such as the title, description, tags, uploader, size and length of the video. Furthermore, there are speech transcripts, visual concept detections and shot segmentation data available. This bunch of data for 11482 videos in total, needs to be indexed and saved somewhere so it's available for the system to search and use.

The next step is the actual linking of the anchors to targets. Following the perspective of [9] the system should extract a query representation from the anchor segment: the indexed data from the step before is accessed to find the anchor and processed in order to identify linkable terms or concepts. Building on that information, the system can construct a query to send to the search system.

Finally, following the same perspective, the system should identify and present potential relevant segments. Video hyperlinking could be seen as a retrieval task, albeit without a user-created query [7]. The search system returns a set of videos that are relevant to the involved system-created query based on the anchor. From this result set, precise targets from the videos should be extracted and re-ranked so that the most useful segment is on top of the ranked list, to minimize user effort and increase user satisfaction.

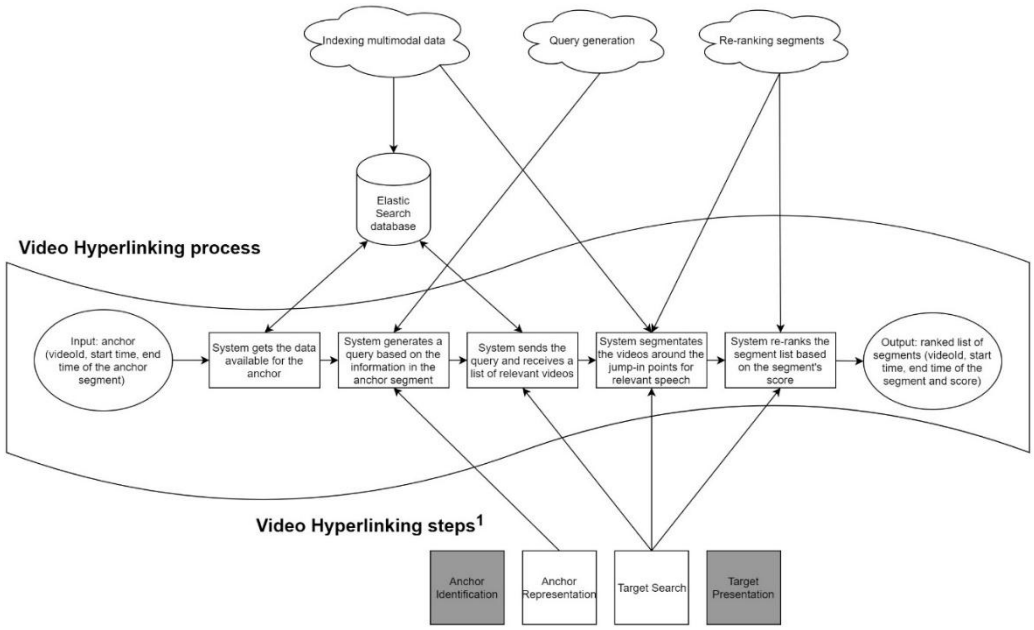


Figure 5: The Video Hyperlinking steps from the definition in Ordelman et al. (2015) and the adapted process flow.

Figure 5 above shows the Video Hyperlinking steps from the definition by Ordelman et al. [9], which are further discussed in 1.3, Background. Above the steps, the complete process from input to output that was used in the proof of concept is given. The arrows from the steps to the process show the relation of the steps with the actual process. In the remainder of this chapter as well as chapter two, three areas regarding the problem of Video Hyperlinking are discussed. The three areas are as well depicted in Figure 5, shown as a cloud. The arrows show the relation of the areas with the actual parts of the process they relate to. The three areas are:

- Indexing multimodal data;
- Query generation;
- Re-ranking segments.

1.2.1. Indexing multimodal data

Video hyperlinking is often seen as an information retrieval task, e.g. retrieve an interesting relevant link given an information need formulated by the anchor segment. The foundation of an information retrieval system is a database, a place where all data is saved so that the system is able to search and work with the data. Databases are systems that hold data in an index. The index allows a search system to efficiently look up the data. The index is built using a specific structure, called a mapping. Because of the complex nature of multimodal data (speech, visual information, metadata), defining a mapping is a difficult task.

When the system is supposed to answer the question “Give me relevant targets about this anchor”, the database where the information resides plays a central role. The database is first accessed by the system to get the content in the anchor segment itself and later to answer the actual query to find relevant content.

There are several important aspects to the choice of database and the difficulty in the accompanying mapping of it. First there is *size.* Because datasets grow rapidly, especially in the multimedia world, the database should be able to handle the load of data coming in in terms of storage, but also in indexing speed. Then there is (search) *speed*, closely related to size, when there is a lot of data to search through, the dataset should be able to search effectively and fast. This is also where the mapping of the data comes in; the mapping should be logical and tailored to the specific usage needs. The mapping defines how "documents" are saved and indexed in the database. But, the multi modal data gives several options on defining a document structure from the data (full video, speech segment, video shot etc.). It is not sure if the document definition stays the same throughout the use of the video hyperlinking system, as new techniques or recognizers are developed (e.g. a face recognition technique that will define a [face + name, start time, end time] pair as a document). Besides, when using a fragment level document definition; it could be that relevant information for an anchor, for which video links are requested, falls outside the specified document or spans across multiple documents. This could result in inaccurate link targets. The problem is thus, how the segmentation can be kept as flexible as possible in order to achieve relevant segments for an anchor.

1.2.2. Query Generation

Information retrieval systems help users find the information that they need. This information need is translated by the user to a query. In video hyperlinking, which could be seen as an information retrieval task, the information need is much less defined [7] because it’s not directly stated by the user but is derived from the anchor segment. We don’t typically know if the user might want to know more about a specific object in the scene or something that was said that he found interesting.

So, instead of typing in text in a search box, a video hyperlinking scenario is acting between the anchor segment that the user currently watches and possible target segments. From the anchor segment, a query has to be generated that will represent the content of the anchor in the search for relevant targets.

Due to the multimodal data and unknown information need, query generation faces a multitude of problems. First, there are different modalities to look for clues. There is the audio (everything that can be heard in the audio) and the visual stream (everything that can be seen in the video). Next to that, the videos also can have static metadata such as a title, description and tags. Secondly,

information in the anchor can be ambiguous. Given an anchor there are multiple interpretations possible, while also the relevancy itself can have multiple interpretations. An example: a program about medieval castles, with in front of the castle a modern Bentley car. There are two distinct entities here: the castle and the Bentley. For the relevancy itself we have multiple options again: more information about the visible castle, shows about other castles, and shows about medieval life in castles etc.

1.2.3. Re-ranking segments

In the developed system in this thesis project, a document-based database will be used (see chapter 3). Therefore, the search engine returns documents that are relevant to the query as a ranked list. Segmentation can be applied before indexing (so documents are already segmented) or after searching (the search engine returns full documents which after the search need to be segmented). Because in this last case the ranking is for full documents instead of the segments, the resulting segments should be re-ranked to reflect the segmentation into the ranking.

Targets relevant to the query should contain just enough context that the user doesn't have to watch a long intro to answer his information need. When having a fixed segmentation, the jump-in point of the target could be off, the actual relevant content could start at the next sentence or shot. Because the ranking is created by the search engine, it should be updated when the segmentation of the results has been finished in order to reflect that some general video could be really relevant, a specific segment from another video could be much more to the point and thus be ranked higher.

The problems occurring in this area are similar to the problems in the query generation subsection. Segmentation of the video can be done on audio (switch in speaker, sentence level) or on what can be seen (shots, scenes) or fixed segments of for example 2 minutes. Re-ranking the segmented videos is another problem, as there are multiple modalities and factors to weigh in, such as keywords in the speech, visual concepts, but also time specific factors such as how long does the user need to wait for the relevant part etc.

1.2.4. Summary

In section 1.2, Statement of the Problem, the problems occurring in video hyperlinking were explained. The first observation is that video hyperlinking is based on traditional text hyperlinking, where anchor texts are linked to pages with additional content. Instead, video fragments are used as anchors. Linking videos by hand is too time consuming because of the large amount of material. Currently websites link videos based on their title, description or other metadata. The aim of video hyperlinking is to provide the user with targets that are *about* the anchor segment, indicating the use of data inside the videos to anchor the links from.

The main problem is: given an anchor X, return a ranked list of relevant targets about X. In order to solve this problem, a process was identified based on the four steps by Ordelman et al. [9]:

- Anchor Identification;
- Anchor Representation;
- Target Search;
- Target Presentation.

The process consists of the following steps (see Figure 5):

First at the Anchor Representation stage:

- System gets the data available for the anchor from the database;
- System generates a query based on the information in the anchor segment;

And then at the Target Search phase:

- System sends the query to and receives a list of relevant videos from the database;
- System segments the videos around the jump-in points for relevant speech;
- System re-ranks the segment list based on the segment score;

From these processes three areas of interest are indicated:

- Regarding the database: indexing multimodal data (also segmentation at indexing time);
- Regarding Anchor Representation: query generation;
- Regarding Target Search: re-ranking segments (also segmentation after query time);

Indexing multimodal data is a difficult task. The mapping of the index should be tailored to the specific needs of the dataset in order to facilitate a quick indexing and quick query time response. Next to that, the structure of the data allows for multiple ways of saving: videos as a whole object or in some kind of segmented form based on speech (sentence level or speech segment), based on objects visible or have a fixed segmentation based on time (2 minutes, 4 minutes). Flexibility in segmentation is an important property because 1) incorporating future techniques and segmentation options (faces visible) is possible and 2) relevant information could span across multiple segments or fall just outside a segment.

In terms of query generation, the problem is in the translation from anchor to query. The users' information need has to be derived from the anchor segment and is not directly stated. The multiple modalities give multiple options to base the query off: speech & audio cues, visual cues and static metadata such as the title and description. Secondly, information in the anchor can be ambiguous and have multiple interpretations.

Finally, the last area of interest: re-ranking segments had similar issues: ranking can be based on multiple interpretations of what is relevant. Similarly to the first area, segmentation is also a problem here. If the videos were not yet segmented at index time, interesting segments have to be extracted from the result set before re-ranking them in the final result list.

The areas identified here will be used as structure in the next Background section as well as chapter 2, the review of current work.

1.3. Background

In this section, the topic of video hyperlinking is further deepened with how the TRECVid benchmark came into existence while also giving an elaborated definition of video hyperlinking. Then each of the three subsections that stated the problem in 1.2, are also used here; giving more background on the stated problems.

One of the first "linking" tasks in a retrieval scenario using multimedia could be seen in the VideoCLEF 2009 track of the Cross-Language Evaluation Forum. The goal of the task was "Finding Related Resources Across Languages" [10]. It ran next to the TRECVid benchmarks, which ran since 2003, but intended to primarily focus on speech and language part of videos. The "Linking" task used a dataset from the archives of Sound&Vision, namely episodes of "Beeldenstorm". The episodes were in Dutch language and the goal was to link them to Wikipedia articles about related subjects, while going beyond a named-entity linking task.

In 2012, MediaEval took upon the "Search and Hyperlinking Task" as a "Brave New Task" [2]- following up on the MediaEval 2011 Rich Speech Retrieval task and the aforementioned VideoCLEF linking task. In the search for "potential new types of user experience", the scenario of the task was

to return known item searches in a video collection, where the known item could not be of enough content to satisfy the information need. Therefore, the search-part of the task was for retrieving the known segment and the hyperlinking-part for retrieving related segments. The MediaEval Search and Hyperlinking tasks ran until 2014, when in 2015 TRECVid adopted the first "Videohyperlinking" task.

The first edition of the TRECVid Videohyperlinking task was motivated by the use case of exploration of a large collection of video data via a link structure at segment level [9]. This link structure can also be seen at Wikipedia, where documents are linked to each other by textual anchors. Distinguishing the use case from a recommendation perspective, the segments should "give more information about the anchor" instead of "give more information similar to the anchor". The anchors for the task were created with a "producer" scenario in mind: participants were asked to enrich a program with video hyperlinks, while keeping in mind the "wikification guidelines[8]".

The increasing use of video online and the growth of visual archives such as at Sound & Vision are the incentive of generating new ways of exploration and discovery of media in the TRECVid research. These new ways are desirable as existing search engines are too limited when it comes to increasing user awareness of the valuable material in archives [5]. Linking can happen in three ways: a) video-to-video inside the collection, where fragments are linked while watching, based on the current segment; b) inside-out, where media from outside the collection is linked to the fragment the user is currently watching and c) outside-in, where outside information is linked to archived media. In this case, video hyperlinking is strictly video-to-video linking. The process of video hyperlinking has been defined using 4 steps [9]:

a) Anchor Identification, anchors in the exploratory phase of TRECVid are triples of video, start and end time, while in the future other variations based on other or multiple modalities could be implemented.
b) Anchor Representation, this step defines a query based on the anchor's data. The multimodal anchor data is processed and relevant information is extracted. The information is transformed into (a/multiple) query(ies).
c) Target Search, the query from step two is applied to a search system that returns a ranked list of results.
d) Target Presentation, the results are presented to the user.

Both the anchor identification step as well as the target presentation step are excluded from this thesis, as anchors are already defined for the benchmark and presentation of the results is not evaluated. In a real-world situation, as is the case at Sound & Vision, these subjects should however definitely be taken into account, by researching, developing and evaluating approaches. The first step, identifying anchors is very important in the real-world scenario, however not included in this work. This step determines the linkable segments from which the user will see links, it could have an impact on the whole user experience if the wrong segments are identified. From the benchmark, each team receives the same anchors and therefore the anchor identification is not performed in the benchmark. The last step, presentation of the targets, is also not evaluated in the benchmark. In real world scenarios, the presentation to the user influences the experience and usability of the system. This work, as it is a laboratory setting with TRECVid data, is focused on step two and three. The Target Search step is split into two areas of interest, first the search system (index) itself and secondly the ranking process that produces the result list.

[8] https://en.wikipedia.org/wiki/Wikipedia:Manual_of_Style/Linking#What_generally_should_be_linked

The rest of this chapter is structured similar to the last section. First, the indexing of the data is discussed (the search system). Then the Anchor Representation step is further elaborated in the Query Generation subsection. Lastly the generation of the ranked list of segments is discussed.

1.3.1. Indexing multimodal data

When looking at databases, the most known form is a SQL-based database. SQL refers to the language “Structured Query Language” which is used to interact with the database. SQL databases are best described as a table with rows. Each row is an entry of data. SQL databases are relational databases – the rows in the tables of data can be linked to each other using IDs. On the other hand, there are the so-called NoSQL databases. In those databases, data is often stored as 'documents'. This kind of database is known for the fast processing of large amounts of data, because it is less strict as an SQL-based database. Still, a NoSQL database needs to know what kind of data is in the database. A mapping file defines the structure of the data that is indexed in the database. Defining a mapping for multi modal video data is difficult because there are multiple streams of data: a) metadata about the whole video, b) data in the speech channel and c) data in the visual channel. Also, defining what exactly a 'document' is, is difficult. A video file would be the most natural way to define a document, as that is the way the data is delivered. All corresponding data such as speech and visual concepts are then saved alongside the document file of the base video. The system could also segment the video, and save the segments as 'documents', as most use cases require some form of segmentation (this is explained shortly). This segmentation can be done on many levels (sentence level, topic level, shot level, fixed 1- or 2-minute segments etc.), and therefore is a problem that is not yet fully researched on what is the best solution - what defines a segment and if it is a more efficient way for indexing is an ongoing discussion.

One could say that there are mostly two groups of users using the archive. One group knows exactly what they are looking for and then there are the others who only have a vague idea [3]. Video hyperlinking could operate for the exploratory uses of the second group, offering serendipitous links to other video material that could be relevant. Because segments are linked together, a segment index would be beneficial. This segment index is also beneficial for the first group that does known-item searches (they might have less benefit from the video hyperlinks though). Query logs show that more and more users want access to video fragments rather than entire programs [11]. But, actually ordering a fragment takes 2.5 times longer due to manually reviewing the material [3].

It is still unclear what form of segmentation is the most ideal for video hyperlinking. Some proposed systems in the TRECVid benchmark use a fixed segmentation, such as Eurecom [12] and Irisa [13] while others use shot-based or topic based segmentation, Polito [14] and FXPal [15] respectively. In general, all systems use some kind of segmentation. A more segmentation-free approach could be beneficial because interesting results could span multiple segments or fall at the end of some proposed segment. While the segment will then be retrieved as relevant, the user still has to watch through the first part of the segment. Next to the segmentation problem, section 1.2.1 also discussed the importance of the underlying database system in regards to the large quantity of data, more especially in terms of speed and the ability to handle large data streams. Among the participating teams in the benchmark [8], Solr/Lucene and Terrier are popular choices because they are built specifically for large datasets and retrieval solutions. Sound & Vision uses Elastic Search, which also uses Lucene under the hood. Elastic Search has interesting features such as Parent-child relationships and term/position payload information but isn't used in the TRECVid video hyperlinking community extensively. The reader is referred to chapter 2 for more information.

1.3.2. Query Generation

Arguably the key challenge in video hyperlinking is query generation because this part identifies linkable concepts and terms in the anchor and creates a query to send to the search system. Therefore, it defines on what basis the links are created. Some "linking" systems use manual links such as Ximpel [16] or are curated by editors, e.g. LinkedTV [17]. Manual links for large archives would be too time consuming. Therefore, automatic systems are being developed [11]. However, they are not perfect yet [18]. Part of the problem is that there isn't a well-defined information need, formulated with a query [7]. Instead, the relevance relation between anchor and targets is based on the system's understanding of the anchor.

The multimodal nature of the anchor segment gives multiple options to generate a query for the search system. Basically, everything that can be seen and heard can be translated into a textual query. This is done by the Informedia team at TRECVid 2016 [19]. They have built a natural language representation layer that represents the modalities in natural language. This has the advantage that known text retrieval solutions can be applied such as TF-IDF. Other teams use a variety of additional sources of information and help such as the GoogleNet deep network used by Eurecom [12], WordNet synonyms to expand the query as used by Polito [14]. The results of the 2016' teams indicate that using the multimodal information could possilbly be beneficial to the retrieval performance. Also, the organizers of the benchmark used anchors in which the speech cues would be referencing what could be seen in the anchor [8]. With part-of-speech tagging in combination with a regular expression, these cues could be detected to find the specific words that are relevant to the anchor.

1.3.3. Re-ranking segments

Recall from section 1.2.3. that a document-based database will be used (unlike a SQL-based one). The search system will return full videos in order to keep flexibility in segmentation (see the implementation details in chapter 3). The benchmark, and users of the system might as well, need interesting segments. Segments will be generated flexibly from the full video based on interesting content from which jump-in points will be set-up. Notice that there are two moments segmentation can happen: at indexing time, the database contains segments which it returns when relevant and after searching, the database returns a full video and the system will segment the result afterwards. Because the topic of segmentation has already been discussed in 1.3.1, it will not be discussed here again.

In 1.2.3. we identified that the ranking by the search engine should be updated when the segmentation has been finished to reflect the fact that while a general video could be relevant, specific segments could be much more to the point, and thus be ranked higher.

There are multiple options to re-rank segments. Functions such as cosine similarity are used by the Irisa [13]and FXPal [15] teams. By using cosine similarity, the relevancy is determined by the angles of the two vector representations of A and B, this is discussed further in Chapter 2, review of current work. Standard TF-IDF (the score of a document is calculated from the Term Frequency and the Inverse Document Frequency) is also used. Weighting some components of the system more than others can influence the ranking scores as well. The Irisa team tested this and concluded that more weight on audio performs worse than on the visual channel, clearly indicating the importance of the visual aspect of this year’s dataset, see further details in chapter 3.

1.3.4. Summary

In section 1.3, Background we delved into the history of the TRECVid benchmark. The video hyperlinking task first starting as a “Brave New Task” at MediaEval 2012, was adopted by TRECVid in

2015. The motivation for the benchmark came from the use case of exploring a large collection of video data, using a link structure at segment level. The increasing use of online video and growth of archives such as at Sound & Vision are real world examples of why new ways of searching and exploring are desirable. The current state is too limiting and therefore leaves users unaware of potentially interesting content.

When looking at indexing multimodal data, we started the section with the most known form of database: SQL-based databases. Data is saved by rows in tables, which can be 'linked' to form a relational database. In contrary, NoSQL databases save data as 'documents'. They are known for their fast processing and ability to handle large data loads, one reason being the loss of the relational model keeping it less strict. Still, the data needs some form of structure, a mapping. There are multiple options for defining a mapping. One could use parent-child relationships (one parent video, with all segmentation forms as child), or use different indexes for each segmentation form. Some systems in the TRECVid benchmark have been using some kind of segmentation at indexing time. Some use a fixed segmentation, others use shot-based or topical segmentations.

Automatic query generation is a very reasonable requirement of a video hyperlinking system. Manual linking or curated links are too time consuming. Some straight forward options such as translating everything to text has been done, while other teams are using additional sources of information such as GoogleNet or synonyms to expand the initial query. Using the multimodal information helps to gain retrieval performance on the 2016 dataset.

Re-ranking segments is in some cases needed, when the database uses a non-segmentation approach. The teams in the 2016 benchmark are using traditional information retrieval techniques such as cosine similarity and TF-IDF to score the results. Weighting components, such as the audio part of the system, is also used. But since the visual aspect of this year's dataset is more pronounced, a multimodal approach performed better.

1.4. Purpose of the Study

The purpose of the study is to implement a video hyperlinking system and learn about the anchor representation and target search steps involved when finding relevant content in a big data collection of multimedia, such as the archive at Sound & Vision, by building a proof of concept using the Blip.tv dataset and TRECVid evaluation methods.

As identified in section 1.2, users of large collections of multimedia are often lost. In case of the Sound & Vision archives, users look for their place of residence and/or some important person that they're interested in and then ask themselves "what now?" [5] [20]. Video hyperlinking offers new techniques to make a web of multimedia content. Use cases are: a) use for exploratory purposes, b) storytelling use or c) for recommendations. Video hyperlinking has a lot of research opportunity, in areas such as anchor identification, segmentation and query generation.

In this thesis in particular, there are three areas of interest: a) indexing multimodal data, its implications on segmentation of video's, b) query generation, what to select from the anchor segment and how to interpret it so that a query is generated that returns useful results tailored to users' needs and c) re-ranking, to try to limit user effort in finding the right targets.

The results of the developed hyperlinking system are evaluated using the TRECVid benchmark guidelines [8]. The TRECVid benchmark of 2016 used the Blip.tv dataset, containing 11482 semiprofessional videos. The participating teams get a list of anchors for which they need to return a ranked list of up to 1000 targets per anchor (called a run). The targets should be between 10 and 120 seconds of length. A team can submit 4 runs for the benchmark. The results from the participating

teams were evaluated using Amazon's MechanicalTurk crowdsourcing platform, where the results were evaluated on relevancy. TRECVid 2016 uses 2 metrics to evaluate the team's runs. The first metric is Mean Average Interpolated Segment Precision (MAiSP) - an adaptation of Mean Average Segment Precision [21], but with fixed-recall points instead of rank levels [22]. The second metric is precision @ 5.

The runs developed for this thesis work are evaluated using the results from TRECVid 2016. The evaluation script (sh_eval [9]) is available to calculate the scores for the runs. For this work, there are 4 runs in addition to a baseline run. The four runs are further discussed in chapter 3.

The thesis is expected to result in a hyperlinking system that presents runs with a score similar to those of the other teams that participated in the benchmark. The expected goal is to gather knowledge about basic linking principals and to turn those into a system that performs reasonably well. From there on further research can be performed on the system to increase the scores as well as to have a system ready that could be used in the next benchmark.

The purpose of the study is therefore:

- Learn about the need for video hyperlinking systems, the large collections and the unavailability of search systems for multimedia result in users being confused and unable to oversee the large quantities of multimedia data.

- Learn about the anchor representation and target search steps in the video hyperlinking process

- Implementing a video hyperlinking system, creating a baseline system and introducing methods to research improvement on the baseline.

- Evaluate the system using the Blip.tv dataset and TRECVid evaluation method (MAiSP and P@5 measures)

- The expectation is that the system presents runs with a score similar to the other teams

1.5. Research Questions

The three subsections of 1.2 and 1.3 give form to the three research questions below:

Q1 How to represent and index the multimodal data so that flexible access to segments is possible?

Q2 What performs better for query generation from speech: using TF-IDF or Part-of-Speech?

Q3 What performs better for re-ranking sub-segments: using selective weighting or cosine similarity?

1.6. Significance to the Field

In this study, a new idea for indexing time-based data such as transcripts was implemented. The idea was devised by Robin Aly through experience with the Axis project [23]. This new indexing technique, used with Elastic Search[10], makes use of the position and offset payloads available for each term. The multimodal data is indexed without any predetermined segmentation, offering flexibility later on, unlike other systems. Furthermore, in this study well-known techniques such as part-of-speech tagging, TF-IDF and Cosine similarity are applied on the new index and further discussed in relation to the self-tuned baseline system.

[9] https://github.com/robinaly/sh_eval
[10] https://github.com/robinaly/videoanalyzer

1.7. Limitations

During this thesis work two issues rose up. Firstly, the mapping that is used to index the data in Elastic Search, is fundamentally different from the mapping used at the B&G archives. Therefore, it is not possible to test the software out on the actual archive content. Secondly, the evaluation of the runs is limited to the results gathered by the crowdsource evaluation of the 2016 TRECVid participants. It could be possible that there are results in this works run that are not seen by the crowdsourcing group and thus labeled irrelevant while they could be indeed relevant. In the discussion chapter a paragraph is written about this with the number of targets that are seen by the crowdsourcing group. The best way to test the actual scores of the runs is to submit them in the following benchmark so the results are taken into account by the crowdsourcing platform.

2. Review of current work

In chapter one, the introduction, the problem was stated. In order to facilitate the user in exploring large multimedia archives, helping them to explore the unknown archive of content beyond simple searches, new ways of exploration are needed. The current searching tools are too limited, especially when the user doesn't necessarily know exactly what he looks for.

The main problem was stated to be: given an anchor X, return a ranked list of relevant targets about x. In this chapter, the problem is further deepened with a review of current work on 3 areas regarding the main problem: indexing (the underlying ability to find and search), query generation (the ability to translate anchor X into a query) and re-ranking (the ability to score and rank results). The source is mostly the TRECVid benchmark; however, also other sources are used. The chapter is closed with a summary.

2.1 Indexing and segmenting multimodal data

In this section the first topic, indexing and segmenting multimodal data, is discussed. The review has three sections: a) review of the concepts that are often used or important to know, b) descriptions of current work at the benchmark of the other teams in this area of video hyperlinking and c) an explanation of the practical implementation of the concepts in this works prototype, as background for the descriptions of the runs in chapter 3 (Method).

2.1.1 Review of concepts

In the background section 1.3., the concepts of SQL & NoSQL databases were introduced. The relational SQL database is saving the data in a tabular form, while the NoSQL form is less strict and saves data in 'documents' – a textual database [24]. Textual databases appear faster for larger data requirements (one reason being that it has less strict relational data requirements) [25]. The model of a text database has three components: the text itself, a specified structure of the text and a query language to query the texts [24]. In Elastic Search, the mapping is the structure that the data needs to adhere to, to be indexed by the search system. When specifying a structure, automatically a certain segmentation is restricted. For simplicity reasons a comparison with books is used here. If one had to index a set of books, the first intuition is to index each book as a 'document'. But also consider other options such as indexing each chapter from each book as a 'document', or maybe each page. Each of these segmentation options also applies to the video hyperlinking problems; one could index each video in full, or apply some segmentation such as speech segments or shot segments. In XML retrieval, this is known as the hierarchical structure of the database [26], each hierarchy has a purpose, for example: logical structure, lay-out structure and results of a part-of-speech tagger. An example of a NoSQL database is Elastic Search[11]. Elastic Search supports a couple of techniques to help with the hierarchy and therefore with setting up segmentation. First is the ability to have multiple indexes. Each layer of granularity could be saved in a separate index. The downside to this is that data is duplicated over these multiple indexes or needs some kind of relational identification. If we use the book example again, these downsides could be explained as follows: if one would like to save books in full but also chapters in order to find some text. In the multiple indexes case, the book's text would be saved in the index Books, but again each chapter's text would be saved in the index Chapters. Al the text is saved twice to keep the ability to find the book title, but also the chapter where one can read the searched text. One obvious improvement

[11] See here: http://elastic.co

would be to not save the full book's text in the Book index. Instead, in the chapter index create an ID to refer to the book where the chapter is in. After finding the text, fire another query to the Book index to find the Book which has the id of the book that's needed. But this way, two queries are needed, increasing the total time needed for the execution. Elastic Search has a novel feature called Parent-Child relationships. This is a way to index chapters as child of a book document. If a query finds the text in a chapter, the query returns the parent document, in this case the Book, as well.

It can be seen now that there are many options to segment and index data. In the next subsection, the work of the participants in the TRECVid benchmark is reviewed on how they have done this task.

2.1.2 Current work

Looking at the 2015 TRECVid benchmark [27], the CMU and DCU teams had the highest scores in respectively the MAP and MAiSP measures. The CMU team defined in their notebook paper a two-step approach for video hyperlinking [28], the first step being segmentation, the second retrieval. They note that the order can be reversed, first retrieval of relevant video's and then extraction of the relevant segments. The DCU team [29] used the latter approach, by using the start time of the sentence as the start time of the target segment. The CMU team used fixed-length segments in their work. Fixed length segments are often used (4 out of the 10 submissions in 2015 and 4 out of 5 in 2016) either because of efficiency reasons but also because the CUNI team reported good results with a 50-second segmentation in the 2014 MediaEval benchmark [30]. The top scoring teams in 2015 and 2016 (CMU, DCU, INF and IRISA) use fixed segments to segment the videos. The Irisa team does report a drawback however, as using this kind of segmentation may cut segments that belong together in several parts (2015, [31]) or in the middle of a sentence (2016, [32]). Some teams do introduce a little bit of flexibility in the segmentation by using the first word of a sentence or after a break as jump-in point or using 'breath' marks as end times for the segments. Finally, there are also teams who use other types of segmentation, such as shot segmentation [14], scene segmentation by using TextTiling (a way to determine scene's in a video) or topic wise segmentation using a TopicTiling algorithm (a way to determine topics) [15].

The Polito team researched the difference between using a fixed segmentation and shot segmentation [14]. They reported the fixed segmentation had a negative impact on the precision at 10 measure, but increased the MAiSP score. This is possibly because the shot segments are all short, while the fixed segments are longer (more of the same shots after each other appearing in the top 10 results, while in the fixed segment case they are all together in one result).

The teams use diverse search systems -from Lucene, Solr (an implementation of Lucene) to the Terrier IR and Gigablast Opensource Search Engine - all systems being of the noSQL kind, optimized for large scale retrieval. Most teams use a stop word removal step and some add stemming as well. The teams do not report any issues with any of the systems.

2.1.3 Implementation in this work

In this work, a novel way of indexing time-encoded data in Elastic Search was implemented. As was mentioned in chapter 1.2, flexibility in the segmentation of the segments that are retrieved is required. This solution was an idea from Robin Aly, who worked for the Axis Project [23]. The basis of this strategy is the use of the payload of terms in the index. Each term has in the backend an offset and position parameter[12]. Regularly, these are used to calculate relevancy based on word positions and offsets. Robin Aly has written an analyzer that at index time, transforms a specifically structured

[12] See the docs: https://www.elastic.co/guide/en/elasticsearch/reference/current/docs-termvectors.html

transcript sentence into terms in the index. The transcript sentence may look like this: “Hello|5|10|World|10|15”. The time codes, respectively start and end times of the words, will be saved as payload with the term in the index. At query time, this information can be accessed to allow going directly to this point in time of the video. This removes the need to define a segmentation (granularity) before indexing, as it can later be determined based on the time codes in the terms. Other, more basic indexing and segmenting strategies were implemented as well to broaden knowledge of Elastic Search and allow for a small comparison (see also image for a clarifying view):

- Video's as a document in a single index;

- Video's and speech segments as separate documents (the segments documents contain all relevant information such as the speech, but also detected concepts in that specific segment), in 2 indices;

- Video's as parent document and speech segments (same content as above) as child documents in a single index;

- Video's as a document in a single index, with time encoded transcript.

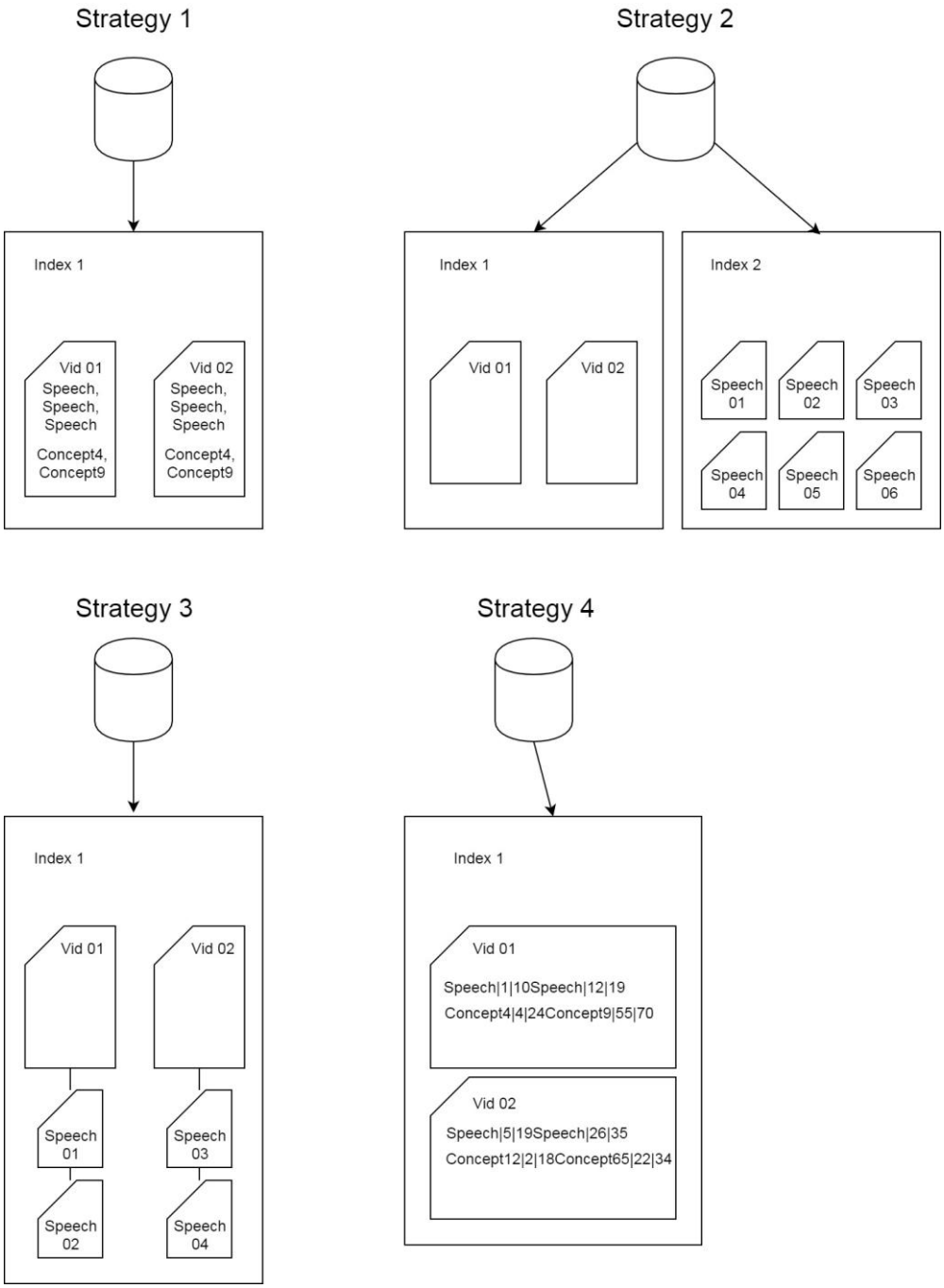


Figure 6: The 4 indexing strategies. Note that the Speech objects (strategy 2 and 3) also contain other data belonging to that segment (concept detection).

Figure 6 shows four strategies. The first one is the most simple. Here, all videos will be indexed as a full document. All data belonging to a certain video is added to the same index in fields. Option two saves two forms of segmentation. The video metadata is saved into one index, while specific speech segmentation objects go into another index. This is also used in strategy three, but here the parent-child option is used. Each speech segment is saved under the video metadata document. Finally, strategy four shows the idea by Robin Aly. Each video will have a speech transcript field, but no specified segmentation. This will be the full audio transcript, in which a segment can be later specified for the user.

Since the last strategy is an unfamiliar and more complex option, it will be explained further. Locked-in segmentations such as the second and third option (using the speech segments as jump-in and out points) are limiting in the sense that if some other technique or segmentation would be tested that another index should be created because it is not possible to add it to the existing one. A practical example would be the times that some 'face' from a face detector to be used as segments. Because the segments are already indexed they cannot be changed. Furthermore, the other problem with fixed segments is, as explained in chapter 1, that relevant content can span across multiple segments, falls outside a segment partly or the relevant content starts at the very end of the selected segment. Thus, a flexible way of segmentation is needed. The problem however, there is a need for a precise jump-in point in the video that can be used to link the user directly to the relevant content. The transcript that is saved as a field in the first strategy does not have any time information saved and therefore it will be difficult to link the user to the specific text segment (in terms of time of the occurrence in the video) that is relevant. In the last strategy, we save the transcript with the word timings inside the string of text. An example would be "hello|3|4 world|5|6". This way, if the system wants to link the user to "world", the system can link to the 5th second in the video. And using this way the flexibility is large, if another field of face detections would need to be saved this can be done by saving "faceID|4|10" etc. Because the segmentation is not indexed, the system can choose the jump-in and out points for each target segment differently and more specialized to the current target.

The indexing of this encoded string of words is not possible in the standard Elastic Search editions. Robin Aly developed a plug-in that enabled it to do so[13]. The plug-in script (miss)-uses the payload functionality. A payload is additional data that is attached to the, in this case, words. Search engines save the position of words in text, the offsets etc in this payload so they are able to use this information in the relevance scoring (words that are positioned closer together most likely belong together etc). The plug-in reads the string of text and saves the start and end time of each word instead of the actual word position and offset. Therefore, when the search returns results, the payload can be used to return the actual jump-in point.

2.2 Query Generation

In this section the second topic, query generation, is discussed. The review has the same structure as before: a) review of the concepts that are often used or important to know, b) descriptions of current work at the benchmark of the other teams in this area of video hyperlinking and c) an explanation of the practical implementation of the concepts in this works prototype, as background for the descriptions of the runs in chapter 3.

2.2.1 Review of concepts

In the introduction chapter, the concept of video hyperlinking was described as an information retrieval problem. After all, it is about answering the question: given an anchor (query) retrieve relevant targets (information). The first concept that an information retrieval problem uses is the so-called information need. Users that use an information retrieval system use it to retrieve information that they are querying for. The user therefore translates his “need” for information into a query that the search system answers. Video hyperlinking is in general the same, instead it is doing multimedia retrieval. Especially: when linking an anchor to targets, the system tries to retrieve targets relevant to the anchor, therefore the anchor is now the query. The process of query generation is the representation of the anchor in the search for relevant targets, by identifying linkable concepts and

[13] It can be found here: https://github.com/robinaly/videoanalyzer

terms in the anchor. Textual databases have three components; the text itself, a structure (2.1) and a query language. The contents need to be translated into the specific query language [24].

In the information retrieval field, the most used technique for weighting and text mining is TF-IDF [33], which stands for Term Frequency – Inverse Document Frequency. Intuitively, if a term is frequently found in a document, it is important for the document. But, there are a lot of basic words such as 'the' and 'and' that also have a large frequency. The Inverse Document Frequency is accounting for that. Words which are frequent in all documents in the whole document-set are less specific than words that have a low document frequency.

Now this technique above works for textual problems, but what else? This brings us to one of the problems in this section: the multi modal nature of the anchors. In order to analyze the anchor segment successfully, more than just text needs to be analyzed, since there are also images. And often times, the image on the screen is in some way linked to the speech that can be heard. The organizers of the benchmark task have also phrased that the anchors are specifically selected on speech cues that are relating speech to the image on the screen [8]. Multi modal approaches to identifying linkable elements are discussed in the next section.

Another textual technique that can be applied to the identification of the speech cues is part-of-speech tagging. POS gives a lot of information about the word and its surrounding context [34]. Using a POS-tagger, complex parse trees can be generated. However, these are not necessary, as partial parsing can have enough information for identifying and classifying information-rich segments in texts. A process of chunking is used to capture interesting parts of phrases, such as noun phrases. Using finite-state rule-based chunking, rules are created by hand for the chunking algorithm. These rules are often regular expressions that capture structures of tagged sentences.

Just using the anchor content itself for the generation of the query is possible, but query expansion [35] is also an often found concept. The anchor can be expanded, in order to let the query reflect context around the anchor. Another way expansion can happen is the usage of for example synonyms, so that more links can be found relating to videos that contain a synonym instead of the actual terms or using a thesaurus [33].

2.2.2 Current work

In the results of the 2015 benchmark versus the results of the 2016 one, there can be seen a difference. The difference is that while in 2015, textual methods excelled at the task; in 2016 the multimodal methods were better. This was due to a change in dataset as well as that the anchors were defined better to be multi modal [8]. A team that experimented extensively with a lot of retrieval methods is CMU-SMU [28]. They've also added context to the experiments to see the impact of context. Based on their experience, content-based methods didn't result in good results in the 2015 task while an adaptation of TF-IDF, namely LemurTF-IDF, performed the best in the 2015 benchmark (MAP metric) and placed second on the MAiSP metric.

Text methods

Not all teams use a query generation component, instead they compare anchor and target pairs directly without searching in the anchor for query terms. In the textual domain this results in some teams using statistic-based methods such as TF-IDF while others use a more bag-of-words strategy with comparisons using vector-based methods. The difference, the TF-IDF based method retrieved results containing the exact terms, while a word2vec implementation introduced noise for the Vireo team [36]. They extracted named entities from each segment and excluded entities based on their document frequency (>10). Each vector representation was further finetuned on the Google News

model. Teams that use a query generation component, for example Cuni and DCU, do so with different techniques. The Cuni team [37] uses all words in a query segment and adds the video metadata to it. They expand the query with context around the anchor and if music is detected, they use a fingerprinting service to detect the title and artist of the song. This information is also added to their query. The best performing team on the MAiSP metric in 2015 was DCU. They use spoken terms as query and expand the query with context [29]. The context is however filtered, only the top terms are added to the query. The weights to determine the best words are calculated with "offer weights". A high weight means that the term has a high document frequency, while having a lower document frequency in the whole collection.

Other teams such as TUZ and IIPWHU also applied textual methods. TUZ [38] generated queries based on subtitles and common words (>10000) are filtered out. The IIPWHU team [39] didn't generate any query, but used the bag-of-words method along with TF-IDF to calculate similarity between anchor and target pairs. They reported that including context was beneficial when anchors were short.

In 2016, most teams did not apply a query generation step in their systems but relied on comparison-based methods. One team, Eurecom Polito applied generation of queries using the ASR script, visual concepts and their OCR method, combining them in one large textual query [14]. They further expanded it with metadata, context and WordNet synonyms. They reported that synonyms did not improve the scores. The other teams applied comparison-based methods. The best performing team (2016) on the P@5 measure, Irisa, applied TF-IDF vectors with Cosine Similarity to obtain a textual score [13]. The Inf team (who performed best on MAiSP and second on P@5), did not apply any specific textual query generation and used the ASR script for the audio track. They do however view video's as a text document using their novel Language-Aided Multimodal Retrieval (LAMAR) framework [19]. They do use textual retrieval methods, such as TF-IDF for the similarity comparisons between anchor and target pairs, with LemurTF-IDF performing best.

Visual methods

In the 2015 benchmark, a number of teams did not experiment with the included visual concepts (DCU, Vireo, Orand), or with slight adjustment: IIPWHU [39] used the WordNet database to get a set of words expressing the concept. The TUZ team calculated SIFT features for each keyframe and concatenated resulting words from the Flickr codebook into a "visual sentence" [38]. Next some teams used various more technical approaches, some working better than others. The CMU team used an Improved Dense Trajectory to get motion features, combined with audio and visual semantic features in their content-based run [28]. It however performed badly and results were not included in the paper. Cuni applied the included visual concepts, only when it occurred in more than 7 keyframes in the segment [37]. They went further in their "Feature Signature" run, where they extended the baseline system with Feature Signatures as visual similarity measure. Feature Signatures calculates an approximation of the distributions of color and texture in images. Therefore it can identify segments with similar backgrounds. Finally, the Irisa team [31] considers a video to be a set of keyframes. The similarity was then calculated by building a concept matrix for the segments and then calculating the dot product.

Multimodal methods

Most teams did not fully leverage the multimodal opportunities in the 2015 benchmark. IIPWHU did a slight throw at it using linearly fusing both modalities [39], however the best weight was 0,2 on visual and 0,8 textual. CMU also used a linear combination, by combining the individual relevance scores of each feature [28]. Combining visual and acoustic features worked best for the Orand team

[40]. Finally, Irisa [13] did the most impressive work regarding multimodality, the cross modal approach is an interesting idea. It works by translating one modality to the other. Their research indicates that the use of multimodal techniques offers better targets, because the visual to audio translation resulted in better performance than the pure visual modality. They also report an important issue, near duplicates were relevant while they shouldn't be evaluated as such, encouraging the design of methods that introduce diversity and serendipity.

In 2016, the task changed along with the recommendations after the 2015 edition, gaining interest for multimodal methods by the teams. The anchors were specifically chosen to reflect multimodal importance. While the Irisa team performed in the lower half of the runs in 2015, their approach resulted in the best score (P@5) and a 3rd place (MAiSP) in the 2016 benchmark [8].

The best score on the MAiSP measure was for the INF team. They [19] applied their Language-Aided Multimodal Retrieval system (LAMAR) to be able to use regular text retrieval methods as all the modalities translated into the same common language space. They use the ASR transcript to extract speech, while using "Frame Content Summarization" on the keyframes. The textual representation of the videos therefore exists of:

- Speech
- Concept detection
- Dense Image Captioning
- Scene OCR
- Metadata (title, tags and description)

They used two models, Terrier IR for the vector space models and the Word2Vec model trained on Google News. The TF-IDF method on Terrier IR performed the best.

The FXPal team's baseline run performed the best out of their runs [15]. It is as well based on a TF-IDF vector based similarity. The multimodal adaptation is a weighting function that combines the results from the different modalities. The weights are however still more leaning towards text: 0,6 text and 0,3 visual.

The Eurecom team [12] used both the included concept detections as well as an GoogleNet deep network variant. They calculated confidence scores that reflect the verbal-visual content connection between anchors and targets, these confidence calculations include the visual features in order to make it multimodal. The Eurecom Polito team applied both modalities in their search system, at query generation time they added the textual concept detection to the query [14].

2.2.3 Implementation in this work

In this work, the focus lies on the textual side, more especially, the transcriptions. The transcripts contain much semantic information which can be utilized to generate a query that can reflect the anchor well. Two strategies for generating word for the query are tested, firstly the more statistical approach using TF-IDF and secondly the language analyzing method, using part-of-speech tagging. Both are also combined. Weighting was used to include the visual concepts in the query. More detailed explanation can be found in chapter 3, were all the specific runs are described.

Using TF-IDF to find the most important words in the transcript text seems very intuitive. From inspection, often keywords are detected, but still it is very unknown if those keywords are the focus of the anchor or if the user is actually interested in those keywords. The part-of-speech tagging approach tries to include the speech cues that the organizers have described in [8], and therefore should be capturing a more focused set of words that are at least relevant to the intent the video

creator wanted to convey in the segment. Still, it is very difficult to know what the user, and not the creator, wants from an anchor. In order to include the visual modality in the query, a simple weighting method was used. Visual concepts that are present in the anchor are selected and added to the query with a weighting component (component*x, where x is the weight). Elastic Search engine uses this weight in the retrieval process.

The process depicted in figure 7 shows the complete query generation algorithm.

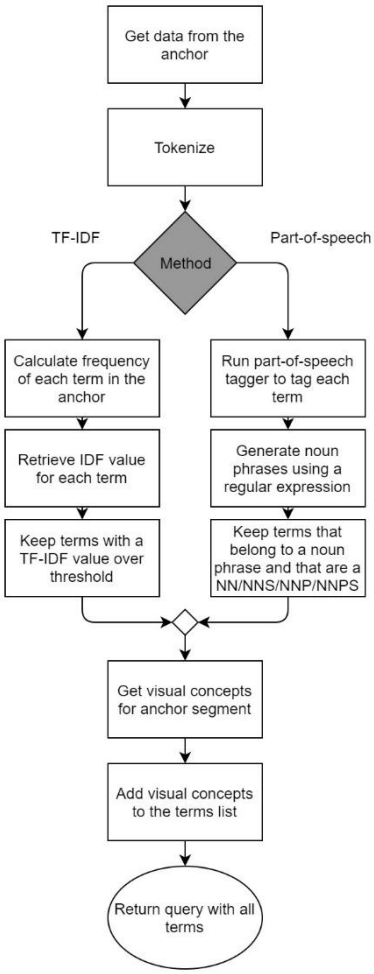


Figure 7: The query generation process used

2.3 Re-ranking

In this section the final topic, re-ranking, is discussed – along with segmentation issues. The review has the same structure as before: a) review of the concepts that are often used or important to know, b) descriptions of current work at the benchmark of the other teams in this area of video hyperlinking and c) an explanation of the practical implementation of the concepts in this works prototype, as background for the descriptions of the runs in chapter 3.

2.3.1 Review of concepts

In order to say something about what relevance is for a system, a similarity measure is designed. This helps to measure the relevancy in a way that is comparable. This could be binary (relevant or not relevant) or could have a score, which turns the system into a ranked retrieval system. Ranking is the process of scoring items in an order of, in this case relevancy, on the basis of some elements in the item.

One simple method to rank segments is to assign weights to important elements in the segment. This introduces two questions, what items should be weighted and secondly, how much weight should be added (how important is this element for the ranking). When there are lots of elements, this can become a tedious and difficult task (imagine experimenting with 10 different weights and checking the impact on the ranking). Luckily, search engines such as Elastic Search return results for a query in a ranked list of results. This however assumes that the results are directly useable for the user. Recall that segmentation can happen at indexing time (the documents that will be returned are already segments) and after querying (the documents are relevant, however not yet segmented). When using a setup that the engine returns segments, the ranking of the engine could be used. When the engine returns full documents, the segments need to be extracted from the results, by determining relevant jump-in points and re-ranking according to weights or using other methods such as the cosine similarity measure.

Another method is the Vector Space Model [33]. Considering the query as vector, a similarity measure is the cosine similarity function, where the cosine of the angle between the vector of both the anchor and the target segments is the score. The vectors represent the data or features in the segments. When

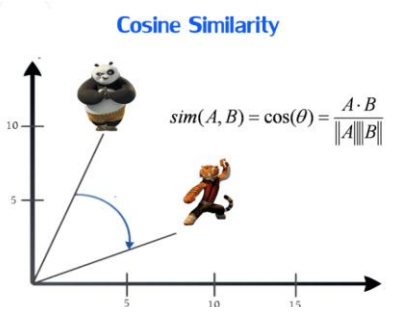


Figure 8: Illustration of Cosine Similarity. Source: Polamuri, S. www.dataaspirant.com/2015/04/11/five-most-popular-similarity-measures-implementation-in-python/

the vectors have an angle close together, the target is similar to the anchor. When the angle is larger, the similarity decreases.

2.3.2 Current work

Using weights to apply more focus on certain elements is a practice that is often used by the teams in the TRECVid benchmark. The actual amounts for those weights differ per team, but in the 2015 edition, the weight was put on the textual elements, while the visual modality was weighted much less in the retrieval. The IIPWHU team [39] found for example 0.1 (title), 0.2 (description), 0.7 (content) to be working the best for their system on the textual side, while applying a linear fusion with weights of 0.2 for visual and 0.8 for the textual score. The CMU-SMU team [28] uses a machine learning algorithm to learn the best weights. For this purpose, they have selected Subtitle data, concept data, trajectory features (motion) and MFCC features (audio) for the calculation. Due to the lack of labeled data, using machine learning appeared not to improve the score for the team. The CUNI team also applied weights [37]. More specifically, in their "series" run, they applied a weight if the target video was from the same TV series. The results of these weights were linearly combined with the search engine results, by using a weight of 35. In the sense of serendipity, giving positive weights to the same TV series might be not something very interesting for users; and the results show that it did not outperform the text-based baseline. They also apply a timing based weight, by using the dates of the target and query anchor. In the "Feature Signature" run, the team applied a combination weight of 90 to combine the scores of the text-based retrieval with the feature similarity retrieval. In general their system works as follows. First a text-based retrieval is started. Then, then those scores are either combined with the feature signature or series run to re-rank the results. The DCU team tried to do something segmentation-free [29]. They searched both video-level and segment-level indexes for targets. Next, both retrieval lists are combined using linear fusion weights. They used a learning algorithm as well to determine the weights, however, just using equal weights performed better - the re-ranking of the initial list actually decreased their hyperlinking performance. For one run, the Vireo team [36] applied a fusion of TF-IDF and weighted word2vec methods to rank the segments. For the weights in the word2vec method, the document frequency is used. Finally, the Irisa team [31] applied a re-ranking method on 2 runs. They built an ngram consisting of unigrams, with weight 2, bigrams with weight 3 and trigrams with weight 5. Then, a convolutional neural network[14] was applied to re-rank the segments.

In the 2016 benchmark, similar setups have been used. The Irisa team for example, computed similarity with the cosine function, while combining the visual and audio modalities using weights of 0.7 for audio and 0.3 for visual features. Their idea to improve on this was to use a bidirectional neural network [13], instead of the CNN from last year. The bidirectional network learns from audio to visual modality and vice versa. The results are then combined using weights, which are learned as well. The Inf team [19], also performing well in the benchmark, explored various weighing models, such as BM25 and TF-IDF. The "LemurTF-IDF" model performed the best. FXPal used dynamic weights for the multimodal re-ranking [15], including topical information and uploader intent in the similarity measure. The uploader intent inclusion has the motivation that the anchor uploader must be the same as the segment uploader or the anchor uploader should have commented on the target video, which could possibly limit interesting results from other uploaders. The writers did not include explanation of why they used this method. The re-ranking from FXPal used the following weights: 0.8

[14] CNN's are networks often used in image & video recognition, recommender systems and natural language processing. Inspired by biological processes, the network consists of 'neurons'. The neurons learn to convert input into output (for example, labeling an image with an object to name the object) [47].

for text, 0.2 for visual when the topical similarity is below a certain threshold and the uploader doesn't match. When the topical similarity is above the threshold, a weight of 0.1 for this similarity is added, while the same weight is also applied to the uploader similarity if they match. The weights of the textual and visual modalities change to 0.6 and 0.3 accordingly. Finally, the Eurecom team [12] also applied weighting to combine the original search engines' result with the additionally calculated confidence scores from the visual concepts (reflecting the connection between the visual concepts and the verbal transcripts). The weights they used were equal for each confidence score (0.2 each), or 0.35 and 0.1 if word2vec was used in the run.

Between the 2015 and 2016 benchmark, the use of vector based models was increased. Unknown exactly why this is, only the Vireo and Irisa team used similarity in terms of vectors in 2015. Vireo used a vector representation of the subtitles, in order to use cosine similarity as ranking the relevant segments. Their word2vec run did not outperform their TF-IDF runs however [36]. Irisa also presented a run where cosine similarity was used as re-ranking measure [31]. The visual-audio representations were re-reranked using ngrams in the cosine calculation and performed better than the other runs. In 2016, most of the teams (4 out of 5) used cosine similarity. The Irisa team used cosine similarity in several of their 2016 runs [13]. Firstly, they used it to compute similarity between the audio representations of the target/anchor pairs for the audio score. Also a visual score was computed using the same method, by calculating a visual concept vector over the whole dataset. Finally, both modalities are compared using cosine distance as similarity measure. The INF team used their LAMAR framework to translate each video into a textual representation. The cosine similarity measure was used to obtain a relevance score [19]. In two of their runs, the Eurecom team also used cosine similarity [12]. It was used to compare the visual vectors with topic vectors. Out of the four runs of the FXPal team, their best submission used cosine similarity to find similar segments [15].

2.3.3 Implementation in this work

In this work, an Elastic Search instance is used that has the videos indexed without segmentation. Therefore, after the results are received from the search engine, jump-in points have to be determined before re-ranking the specific extracted segments.

The jump-in points are based on speech. The first word in the speech transcript that matches a term in the query is selected as the ‘start’ point. However, since this could return segments that start mid-sentence, a slight bit of context is used so that the segment starts at least a bit before the relevant matching word. Punctuation cannot be used to find the first word in the sentence to mark as the jump-in point, because the punctuation was removed in the tokenization step. A future update should address this.

Next, the following matching term in the transcript is analyzed. If this term is close in terms of time to the word before, the segment is expanded to include this word. More specifically, if the next matching term falls inside a window of 4 seconds after the last matching term, the segment is expanded and the window is moved to this term. If the term is outside the window, a new segment is created and the one before is closed. The length of the window was decided after testing different values on the development set.

There are two methods used for the similarity measure & re-ranking phase in the project. The two methods that have been experimented with have also been used by the other teams: a weighting function and vector-based cosine-similarity. For the weights the following set was used: the original Elastic Search relevance score was divided by 10, to reduce its weight. Then when a word occurs in the query as well as in the segment, a weight of 4 is added to the score. Then each time the segment gets enlarged because another word fits in the window, again, the score is enlarged with 4. So, the

more words are added (length of segment) the higher the score. Next, diversity of words: if the new word is not yet in the segment, the score is multiplied by 2, otherwise just added 1 point. If one 'keyword' is often said, it doesn't get the segment a too high influence in the score. More diversity in the words is better for the score in our baseline weight system. If a visual concept is found in the segment, which also is in the query, the score is also multiplied by two, in order to give the visual modality influence to the ranking. All these weights have been experimented with using the development set of anchors to tune them.

The second similarity measure that was used in the project is the cosine similarity function. Both the transcripts of the query as well as the target's transcript were vectorized (by creating an array of the TF-IDF values of the words). Next, the relevance score was calculated using the Python cosine similarity function, taken from dataaspirant.com[15].

2.4 Summary

Given the problem: return a ranked list of relevant targets about an anchor X, the review of the current work was divided in three similar sections to the structure from the Introduction chapter (indexing multimodal data, query generation and re-ranking targets). For each of the sections, the work was divided into three sub sections: review of important concepts, review of current work and review of the concepts used in the project belonging to this thesis work.

First the relating concepts to indexing and search functionality were discussed. The requirements to a large-scale data system translate into the use of 'document' based databases instead of the tabular formed SQL databases. Examples such as Elastic Search and Solr are providing functionality for indexing of and querying for data. These search system, although much less strict, still need to know about the data that is to be indexed. Several indexing strategies were implemented:

- Using one index for all data
- Using multiple indexes for each modality or segmentation form
- Using parent-child functionality, let the segmentation form be a child of the parent base form
- Using one index, but using time-coded strings for each modality in order to save the time aspect instead of locking segmentation at indexing time.

The last option is a novel technique and not used before and allows the system to create segments in a flexible manner. The segmentation free methods can solve drawbacks like cutting of the video in between parts that should belong together. Some teams are slightly flexible by using sentence structure or breaths as cutting points for segments. In this work, Elastic Search is used to employ the payloads associated with terms in the index. Using a plugin by Robin Aly, it was possible to, at index time, save the accompanying timing information encoded with the transcript. The start and end times of the words are saved as payload and allow the system to do segmentation flexibly at query time.

In order to achieve an understanding of the content of the (query) anchor, two methods of text mining have been discussed. Firstly, the statistic based TF-IDF function – calculation of the term frequency in relation to the frequency the term is in the number of documents in the whole collection in order to say something about the specificity of that term for a document; secondly a language modeling method, called part-of-speech tagging, in combination with a chunking process in

[15] See here: http://dataaspirant.com/2015/04/11/five-most-popular-similarity-measures-implementation-in-python/

order to classify segments in texts that contain information. Rules are created by hand for the chunking algorithm to capture specific structures of text.

In 2015, the TF-IDF method based on mostly text has been utilized extensively, where in 2016 more sophisticated methods came into play in order to adhere to the more multimodal nature of the task. This also reduced the use of a query generation component, as teams are comparing anchor and target pairs directly on relevancy. The LAMAR framework from INF team performed best in the 2016 task. The textual methods still work for their framework because each modality is translated to text first. Irisa developed a cross modal approach by translating the audio modality into the visual modality and vice versa and combining the results together, returning a top score in the Precision @ 5 measure in 2016. In the query generation department, this work focusses on text methods only (the visual concepts are used, but don't play the main role in the system). This is to develop a baseline system. Both the mentioned TF-IDF and part-of-speech methods will be implemented to compare the results.

Comparing anchors and targets is should be based on 'relevance', although most computational methods are similarity measures, which may not directly saying something about relevancy. There are two methods to calculate a score to reflect relevancy in order to rank the results. The first is weighting. This tedious job can include multiple elements that can be identified to be important for the ranking score. Each of the elements has a weight that is tuned on the development set. This practice is often used by the teams in the benchmark. In 2015 the teams focused their weights on the textual side of their systems, while in 2016 more vector based methods were used, the teams adopted cosine similarity to use as another method of scoring similar segments.

3. Method

3.1 Introduction

This chapter describes the study conducted in this work. In this study, the development of a video hyperlinking system is discussed. Searching in video's and exploring large collections of video's is an emerging problem since online video is becoming a largely used platform for entertainment and learning purposes. Existing search and linking happens in a text-based query manner. The aim of a video hyperlinking system is letting the user explore videos using linking on video segment basis. When building such system, there are several areas of interest. The following problems are reviewed: a) indexing multimodal data and the influence on size, speed and segmentation issues; b) generation of queries from an anchor segment, selecting the right content from the anchor is essential to getting relevant targets and c) the re-ranking of returned documents, while full documents in total could be relevant to the suggested query, it could be that some segment from another video is much more concise and in general more fitting to the query.

This chapter describes the method used to answer the following research questions which were stated in the introduction:

How to represent and index the multimodal data so that time-code access to segments is possible?

What performs better for query generation from speech: using TF-IDF or Part-of-Speech?

What performs better for re-ranking sub-segments: using weighting or cosine similarity?

The chapter is structured as follows: first, a description about the TRECVid Video Hyperlinking benchmark task, goal and restrictions are given. The next section describes the dataset; section 4 describes the implementation of the basis of the system. Section 5 gives descriptions about the runs proposed in this work, continued by section 6 where the metrics are reviewed. Finally, the last section gives an analysis of the metrics and what their influence is on the results.

3.2 The TRECVid Video Hyperlinking benchmark

The TRECVid Video Hyperlinking benchmark task brings researchers together to work collectively on the subject. The 2016 edition [8] had 5 participating teams: Irisa, Inf, FXPal, Eurecom and Eurecom Polito. They submitted 4 runs each to the task organizers. The tasks' goal is to return a ranked list of targets per anchor. There are several restrictions about the targets properties:

- A target segment should be *about*[16] the anchor segment.
- A target segment should not come from the anchor video.
- A target segment should be between 10 and 120 seconds long.
- A target segment should not overlap with other targets for the same anchor.

3.3 Dataset (Blip.TV)

The 2016 edition of the Video Hyperlinking task used the BlipTV10000 dataset [41]. In the dataset are 14838 semiprofessionally created videos with a total length of 3288 hours. The videos are together with their metadata crawled from the website Blip.tv. The publishers of the dataset have also added automatic speech recognition (ASR) transcripts and shot boundary detection results. The 2016 edition of the task added new versions of the ASR transcripts as well as visual features from keyframes within the shots. The visual features available in the dataset are classified using the ImageNet dataset trained using deep learning techniques. The deep convolutional neural networks

[16] About in the sense that the target should be related to and not necessarily similar to the anchor.

were trained by Jeff Donahue using a minor variation on CaffeNet implementation of Alexnet, see for more information ([42] and caffe.berkeleyvision.org). This work uses the new 2016 version ASR transcripts provided by LIMSI which are based on neural network acoustic models.

3.3.1 Development & Test set

From the dataset, a development and test set of anchors are given by the organizers [7]. The development set consists of 28 anchors, defined by crowd sourcing for the Search and Hyperlinking task in 2012. The test set contains 98 anchors. These were created by searching the dataset for spoken cues in the transcripts. Examples of those cues include “can see”, “this looks” etc, all linking the spoken modality to the visual modality. All anchors from the development and test sets were verified by an Anchor Verification assignment on the Amazon Mechanical Turk platform.

3.4 Implementation

In order to build a system for video hyperlinking, the first look is to the development of the database and index. Flexibility of segmentation was achieved by not using a predefined segmentation at indexing time, by using the analyzer plug-in from Robin Aly[17] - each video is saved as a document with a field containing a time-encoded speech transcript string. To be able to index the dataset into Elastic Search, all the data from the Blip.TV dataset was transformed to JSON and the speech transcripts were transformed to a string with the time data encoded in the form of ex: “Hello|12|14 World|15|19”. A broader knowledge of Elastic Search was gained by also implementing the other strategies mentioned in chapter 2, based on multi-index and parent-child indexing. Because of the flexibility requirement, the index that used the preprocessed data with the analyzer plug-in was used to build the runs on that are described in the next section.

3.5 Description of the runs

For more background information around the implementation, read the “Implementation” sections in chapter 2.

3.5.1 Generating segments

Because the search system returns full videos, the results presented to the users are segmented interactively from the result set using a 4 second window based on the transcript. The first word in the transcript that is in the query is the start of the segment. Each word that fits in the window is added to the segment and the window is moved to this word's time until the total length of the segment reaches its max of 120 seconds. If the next word in the transcript that matches the query is outside the 4 second window, a new segment is created.

3.5.2 Run 1

The first run uses Elastic Search to boost certain words in the query. The keyword selection for the query generation is based on tokenization and part of speech tagging. NP, NNP and NNS tagged words are considered keywords, which are boosted by a weight of 4 (if the word is found more often, the weight is lowered by 1, until 0 – if the word is occurring 4 times it is not a very distinctive word). The segments are re-ranked using the following weights[18]: 0,1 * documentscore (this is the Elastic Search score of the whole video). For each word that is added when found inside the interactive segment window the score gets +4. Diversity of words is important, therefore, if a new keyword is found in both the query and the target, the score is multiplied by 2, if the word was said before just one point is added (so repeating the word doesn’t influence too much). If a concept is relevant (is a

[17] View on Github: https://github.com/robinaly/videoanalyzer
[18] The weights were tuned using the development anchor set.

match) for the segment it gets +1. Each matching concept multiplies the score by 2. The weights are tuned on the development set of anchors to achieve best results.

An example query for the following transcript of anchor 16:

Transcript: "{fw} . This bill would send a strong message to domestic abusers and what would be potential abuses abusers that domestic abuse will not be tolerated. That their violent past will not go unnoticed"

Query (transcript part only): "bill^4 message^4 abusers^3 abuses^4 abusers^3 abuse^4 violent^4 past^4"

Words that describe the NPs, such as "domestic" in the case above, are not captured.

3.5.3 Run 2

In run 2, a more advanced version of the part of speech tagger is used to test improvements by using regular expression grammar to catch keywords. The generated noun phrases are added to the query string. Patterns were tested manually, for example incorporating the use of the <IN> preposition tag or the <DT> determiner tag. The TRECVid organizers have specifically said that the anchors are based on speech cues, which could be caught using a grammar (for example: can see, seeing here, looks like, showing etc.). The grammar that is used as the rule for the chunker of the part-of-speech tagger is as follows (the tags are found in the Penn Treebank[19]:

```
NP:   {<NN>*<IN><DT>*<JJ>*<NN|NNS>*<CC>*<DT>*<NN|NNS>+}
      {<TO><VB>*<DT>*<JJ|JJR>*<NN|NNS>+}
      {<NN|NNS>*<VB|VBP|VBN|VBD|VBZ><DT>*<JJ>*<PRP\$>*<CD>*<NN|NNS>+<WDT
      >*<JJ>*(<,>*<NN|NNS>)*}
```

Words that are found in an NN/NNS/NNP tree are labeled as keywords. In addition, if a word is either VBP or JJ it is also added to the query (if it is not a stop word).

An example query for the following transcript of anchor 16:

Transcript: "{fw} . This bill would send a strong message to domestic abusers and what would be potential abuses abusers that domestic abuse will not be tolerated. That their violent past will not go unnoticed"

Query (transcript part only): "send strong message domestic abusers potential abuses abusers domestic abuse tolerated violent unnoticed"

As can be seen, now, "domestic" is captured in the query.

3.5.4 Run 3

The third run uses the statistically based method TF-IDF as implementation to test if that produces good results. The term statistics are used to calculate the TF-IDF value of each word in the transcript. A threshold of 3.45 was found to divide keywords and non-keywords, again by tuning on the development set. In this run, the cosine similarity function is implemented to test another re-ranking strategy. By utilizing the creation of the vectors at the query stage, the query vector can easily be compared with the segment vector for each result.

An example query for the following transcript of anchor 16:

[19] See here: https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html

Transcript: “{fw} . This bill would send a strong message to domestic abusers and what would be potential abuses abusers that domestic abuse will not be tolerated. That their violent past will not go unnoticed”

Query (transcript part only): “bill^1 send^1 strong^1 message^1 domestic^2 abusers^2 potential^1 abuses^1 abusers^2 domestic^2 abuse^1 violent^1”

This method based on TF-IDF captures the important words as well.

3.5.5 Run 4

This run combines both the part-of-speech tagging query generation method from run 2 with the TF-IDF calculation from run 3. So, only words that have a TF-IDF value over 3 and are NN/NNS/NNP/VBP/JJ are added to the query (because the part-of-speech tagging with chunking already removes noise terms, the threshold could be loosened a bit). The cosine similarity function from 3 is used again for determining the re-ranking score.

Transcript: “{fw} . This bill would send a strong message to domestic abusers and what would be potential abuses abusers that domestic abuse will not be tolerated. That their violent past will not go unnoticed”

Query (transcript part only): “bill^1 send^1 strong^1 message^1 domestic^2 abusers^2 potential^1 abuses^1 abusers^2 domestic^2 abuse^1 tolerated^1 violent^1 unnoticed^1”

When combining the methods, “tolerated” and “unnoticed” were added to the query.

3.6 Metrics

The following section describes the metrics involving the evaluation of the video hyperlinking system. For quantitative analysis, the first metric is Precision at 5. The second is the Mean Average Interpolated Segment Precision (MAiSP). The metrics are implemented in the sh_eval script by Robin Aly[20].

Measuring performance is a difficult task in an area where an information need is not formally specified using a query and the multi modal nature of the audiovisual data is generating ambiguous needs. It also is still unclear what users actually want from an automated linking system, for example: detail-on-demand versus more contextual links [43].

In the TRECVid benchmark, relevance judgments are collected on the results of all the submitted systems. This task ran on the Amazon Mechanical Turk platform. Judgments were collected for the top 5 results for all anchors for all runs. The ground truth was generated using two tasks, called 'Target Vetting' and 'Video-to-Video Relevance Analysis'. In the first task the crowd workers had to watch a target segment and had to select one out of 5 descriptions. One of those descriptions was from the anchor that the target was retrieved for. In case the anchor's description was chosen, the target was labeled relevant. For each target 3 judgments were collected, where the majority was used as a final decision. In the second task, the crowd workers were given the task to give a textual description of what made the target relevant to the anchor [8].

The evaluation script by the TRECVid organizers has been used to measure the results of the four runs quantitatively. In particular, two measures have been used – Precision at 5 and MAisP.

[20] See here: https://github.com/robinaly/sh_eval

3.6.1 Precision at 5

The metric reflects the quality of the top-ranked results that were assessed, due to limited resources, top 5 instead of 10 [8].

The precision at 5 measure shows the number of relevant targets in the top 5 for an anchor. The precision value of all anchors is then averaged over all anchors in the test set. It is important to have a high precision at 5 values, because users often only look at the first few results. It is therefore crucial to include highly relevant results in the top.

3.6.2 MAiSP

Whether the relevant content is retrieved up to rank 1000 in the list [8], enabling comparison between runs below rank 5 in terms of user effort measured in the amount of time that needs to be spent to access relevant content.

The precision at 5 measure does not take into account any of the complications of a multimedia retrieval scenario. It is simply either relevant or not. However, there is much more to it such as the start and end time of the fragment, for example in terms of precision of retrieved segments in a larger document and the distance of the beginning of the retrieved segment to the start of the relevant content [21].

Therefore, a modification of MAP (Mean Average Precision), called Mean Average Segment Precision, was introduced that measured both the rank of the segment and the segmentation quality. The goal was to get relevant segments at the top of the result list and simultaneously perfectly fitted over the relevant content. The MAiSP measure builds upon this but differs from it because the rank levels are exchanged for fixed-recall[21] points. Since Precision at 5 measures the performance at top-5 level, this measure goes further down the list, up to rank 1000. Instead, this metric measures user effort in terms of the time spend auditioning the content for relevancy and the number of seconds of relevant content users can watch starting from the suggested jump-in point [22]. The formula can be found in [21].

3.7 Analysis

Because the benchmark has been finished, it is not possible to submit the runs for relevance judgment. But, the results are comparable with the other team's results which have submitted. Keep in mind that results in this' study's runs could therefore be marked not relevant; however they might have not been seen by the crowd workers and thus marked not relevant, lowering the final score. Furthermore, the prototyping nature of the systems can only give an indication of performance and needs proving in real-world use and multiple other data sets [3].

[21] In IR, recall is the faction of the relevant documents that are successfully retrieved.

4. Results

This chapter describes the results of the study; the discussion follows in the next chapter. In this study a baseline video hyperlinking system is developed. Due to increasing use and size of video collections, existing search tools offer limited use in exploring and finding relevant content. The linking happens between so-called anchors, the origin of a link, to targets. The linking helps users to get relevant content based on the content they are watching currently, in order to broaden their use of the collection. The content of the anchor is analyzed in order to provide the user with targets relevant to the current need. This study has been structured along three main topics, indexing multimodal data, query generation and re-ranking. The study followed the method outlined in chapter 3. First, the database/index was developed and the dataset was transformed to adhere to the mapping and inserted. The indexer uses the plug-in from Robin Aly in order to index a transcript string containing time-based data and saves it in the payload of each term. This enabled flexible segmentation, as each word can now be the start of a new segment. For comparison reasons, other regular indexing strategies were also implemented, but the main point of the study was using this novel idea by Robin Aly.

The study continued with the development of the specific runs to answer the query generation and re-ranking questions quantitatively using the evaluation script that calculates measures such as Precision @ 5 and the Mean Average interpolated Segment Precision.

In order to prevent over-fitting of the system, the development-set of anchors was used as a model to build the system on. The desired output is known and therefore can be used to make decisions and tune variables on to maximize the scores. This way, the goal is that the performance of the system can be predicted. For the actual results, the system is performing the same algorithms now on a test set, that contains similar data, but not yet seen and therefore the parameters could not have been optimized for this test set (which would be undesirable).

The BlipTV dataset contains 14838 semiprofessional user-generated videos. From this dataset, 122 anchors were created. The development set contained 28 of them, the rest were for the test set.

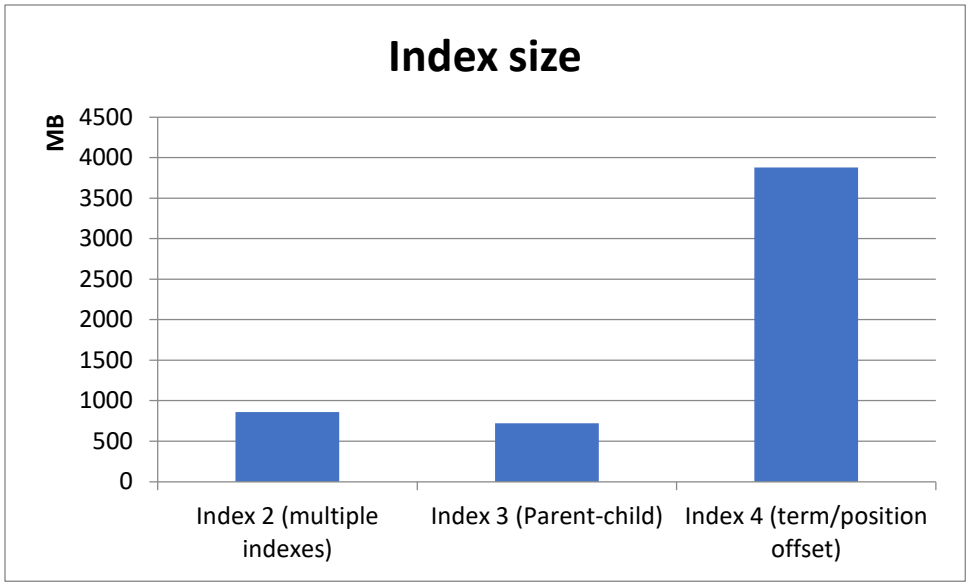
4.1 Implementation results

Robin Aly's technique of using the term's position and offset payloads to save time-based data along with the actual term is a new way of indexing multimodal data. The problem with indexing these kind of multimedia data is segmentation. Each definition of segmentation can possibly limit the effectiveness of a target segment. For example, the relevant content can span across multiple segments or start at the middle of a certain segment. Using this novel way of indexing, this time data can be accessed from the results and be used to generate segments on the fly.

4.1.1 Indexing multimodal data

For comparison sake, two other strategies of indexing were also implemented. Those have the issue that they are not flexible in segmentation and therefore have not been used further. Speech segments have been used as segmentation in those indices.

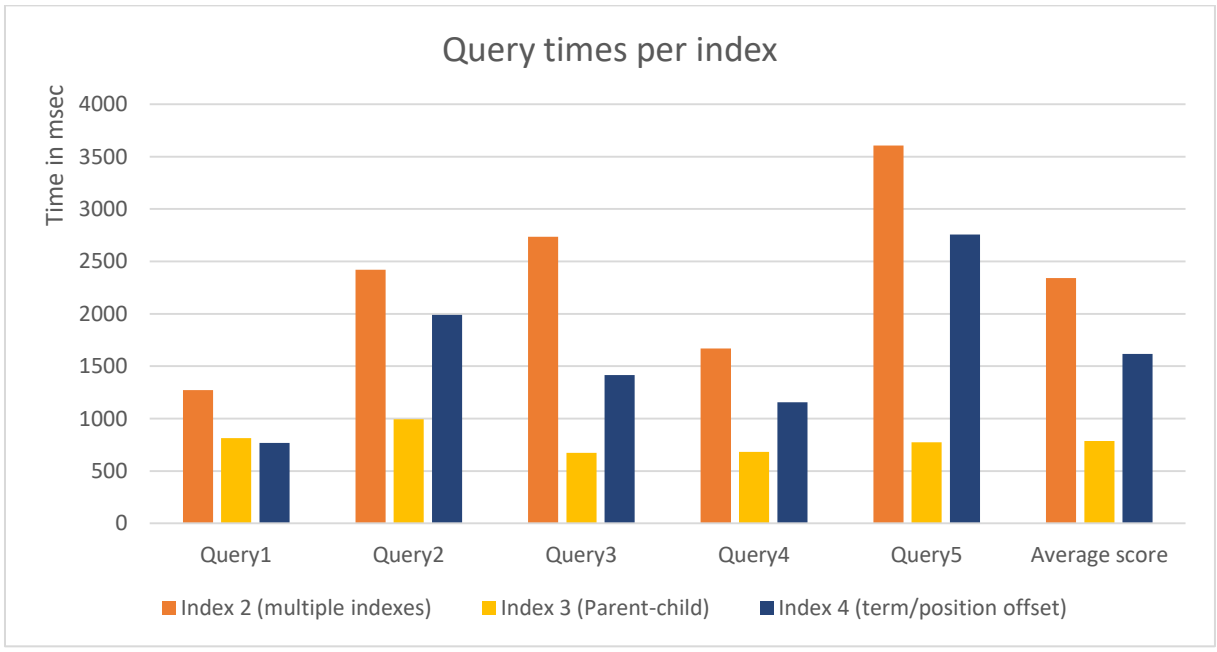
Some data was recorded for the comparison, the size and query times of the indexes have been noted. The 'everything-in-one' large index (strategy 1 from Figure 6, chapter 2) was not implemented as it is very similar to the index with the analyzer script (using the term position/offset).



Dividing the data over multiple indexes slightly increases the index size. The biggest size jump is for the term/position offset index. It can be clearly seen that all the extra timing data costs a lot more space on the disk.

Also the query speed was compared with some simple textual queries, the following queries were send to the indices:

- Trafalgar (a word that is only in a small set of documents)
- My (a word that is in a lot of documents)
- Cooking (an object/concept)
- Spoon (an object/concept)
- How are you (a sentence, multiple words)



It can be seen that the Parent-child index is fairly stable throughout the queries. Using 2 indices and combining the searches takes most of the time. The term/position offset index is performing averagely at query time.

4.2 Quantitative results

Recall the following four runs:

- Run 1, Baseline, weighting keywords by boosting them

- Run 2, Using the part-of-speech tagger to capture speech ques.

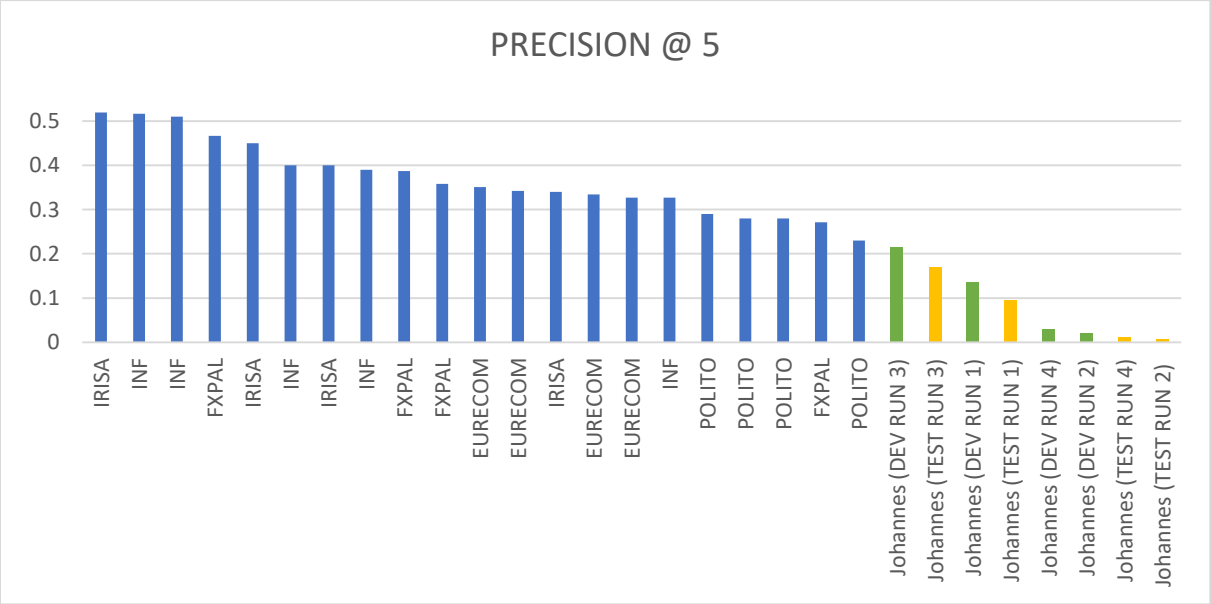
- Run 3, TF-IDF method, threshold, >3.45 is a keyword - cosine similarity re-ranking

- Run 4, Combining part-of-speech to capture speech ques with TF-IDF (only keywords > 3 in order to reduce noise) - again, cosine similarity re-ranking.

Each run was evaluated with the official test set of the anchors and through the evaluation script from TRECVid. The results are further discussed in Chapter 5, the discussion and conclusion.

4.2.1 Precision @ 5

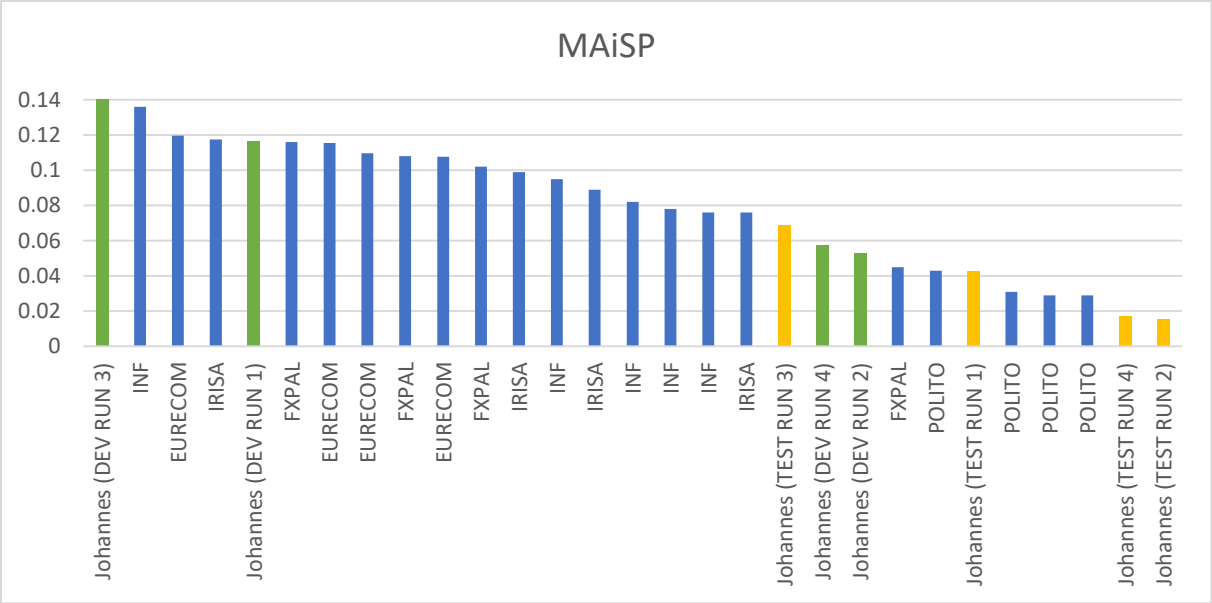
The precision at 5 measure shows how many of the top 5 results are relevant. The values here are averaged over all the anchors.



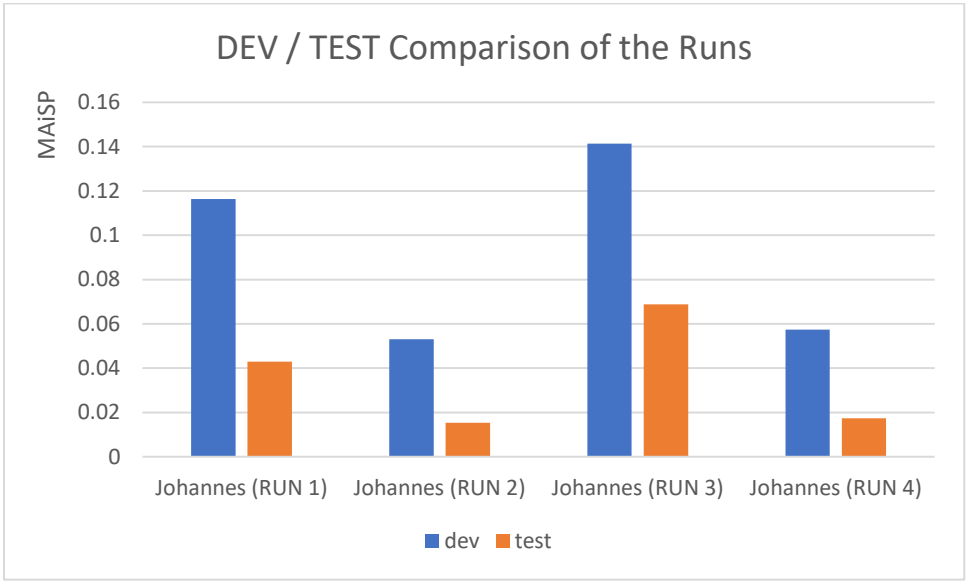
The precision @ 5 measure shows that the number of relevant retrieved results are below what the other teams produced. Especially the part-of-speech results are very low.

4.2.2 MAiSP

The MAiSP measure shows the user effort, in terms of time that needs to be spent checking for the relevant content. The binary relevant-not relevant decision of the precision at 5 measure is extended by instead counting the distance of the retrieved content in relation with the start of the actual relevant content.



A large difference between the expected development results and the actual test set results is visible. Where the development results are on par with the other participants, the official results are ranked low. Similar to the Precision @ 5 measure, run 3 (the statistic based TF-IDF run) is the best performing run.



The difference between the expected development results versus the actual test results is very large.

5. Discussion

The results for this study are discussed in this chapter. A limiting search tool results in users being unable to find content in- and explore large collections of video. The task of video hyperlinking is to aid users in exploring collections by offering in-video links to targets that are “about” the (anchor) segment the user is currently watching. For this task, three sub problems have been identified: indexing data, generating queries based on the anchor content and finally re-ranking the resulting targets.

For the indexing part we use the full video as a document and deal with selecting a jump-in point later. For the selection of jump-in points, an adaptation was made to the analyzer of Elastic Search. The analyzer can parse a string of text with time information between the words in order to give the possibility that later in code, the right time information of each word can be selected. This way, each word can become a jump-in point and it is therefore very flexible for segmentation purposes.

For the other two processes, query generation and re-ranking targets, techniques such as TF-IDF, part-of-speech tagging, weighting and cosine similarity were developed and tested in 4 runs that were executed against the BlipTV dataset and evaluated using the Precision at 5 and MAiSP measures.

This chapter starts with a discussion of the results. In the next section the limitations are discussed, which are translated into recommendations for future work in 5.3. The chapter is closed with the final conclusions.

5.1 Discussion

5.1.1 Indexing multimodal data

The indexing strategy which uses the adapted analyzer for parsing the time data in the text strings appears to work sufficiently and allows for the flexible creation of segments. Using the comparison in chapter 4, it could be seen that the processing of queries takes a little, 0.83 seconds averagely, longer (probably by more processing time needed to extract the time data from the transcript) and it also uses much more space (537% more than a parent-child based index). It must be noted that not a lot of time was spent on optimizing the queries and therefore the index could possibly be improved. The index requires a lot of space in comparison. However, hard drive space is quite cheap these days, while it also has to be noted that the size is not extremely large (3.8 GB, while drives have an average size of 1000 GB these days[22]). The dataset is 3288 hours of video. If, for example the dataset would grow to 1 million hours of video, the data footprint grows to 1.2 TB (1200GB). If you have the data structure available to save 1 million hours of video data, 1200GB more space isn’t really a big issue. The indexing technique using parent-child relationships (strategy 3) makes sense from user and development standpoint (a video is a set of segments, so the index contains video's, which have multiple segment-children). It also has a smaller data footprint as well as it appears faster when comparing the implementations as shown in chapter 4. However, the requirement that flexible access to the segments is needed, requires us to index a flexible segmentation. But in order to index child segments, a strict form of segmentation is needed which is a large drawback when it comes to flexibility. Strategy 4, which incorporates the time information of each word in the results, allows

[22] "Average capacity of Seagate and Western Digital hard disk drives (HDDs) worldwide from 3rd quarter 2011 to 1st quarter 2016 (in gigabytes per drive)," [Online]. Available: https://www.statista.com/statistics/751847/worldwide-seagate-western-digital-average-hard-drive-capacity/. [Accessed 27 11 2017]

developers to use this time information as jump-in points for segments. Each term becomes a possible start of a segment and this method is therefore a lot more flexible.

5.1.2 Generating queries and ranking results

For the video hyperlinking system, four runs were developed as follows:

- Run 1, weighting keywords by boosting them

- Run 2, part-of-speech tagger used to capture speech ques

- Run 3, TF-IDF threshold, >3.45 is a keyword, with cosine similarity as re-ranking method

- Run 4, combining part-of-speech to capture speech ques with TF-IDF (only keywords > 3 in order to reduce noise), also with cosine similarity as re-ranking method.

For the precision at 5 measure the results for all runs were lower than the other participants of the TRECVid benchmark. The participants reached between 23% to 52% of the top 5 results that were marked as relevant. The best scoring development run of this work reached 21%, while the best scoring run on the test set achieved 17% relevant results in the top 5. Also, for runs 2 and 4, the performance of the system was much lower than expected – the runs show a large difference with the other 2 runs, as they only achieve a percentage of 2% relevant. The best performing run was run 3, using the TF-IDF statistics to select keywords for query generation and cosine-similarity for re-ranking – it achieved 17% relevant results in the top 5. While the TF-IDF calculations have been used a lot in information retrieval systems, and therefore have been proven to work, the results indicate also that using TF-IDF could be better than trying to understand text by applying the part-of-speech tagger. Run 1, the baseline keyword-based system is performing second best (scoring 10%), since words with a high TF-IDF value are often keywords, this result is understandable. Cosine similarity used for re-ranking appears to slightly outperform the basic weighting model; however, this could also be due to inefficient weights – by over tuning or tuning weights on a not exactly representative development set, as will be explained later.

The results of the Mean Average interpolated Segment Precision show a large difference between the development runs and the evaluation results. The results of this measure are for every run more than 50% lower in the test scenario. Where two runs had results across the of top participants at the MAiSP measure (11% to 14%), the actual test results are in the lower region and both run 2 and 4 show a very low result (1.5%). Again, best performing is run 3 (7%), the TF-IDF calculations appear to work the best when evaluating the user effort, while run 1 is second best again (4%).

Several problems may have resulted in the low rankings of the runs in regards to the other participants in the benchmark. Firstly, there is the possibility that the anchors from the development set and test set were too different. This could have influenced the weighting process, as weights were tuned on one set, and could have not been ideal for the other set. Another possible issue is that the weights have been over-tuned. This means that the weights are too specifically matched on development anchors and result in lower scores for the other scenario. These problems are further discussed below.

There are differences between the development and test anchors visible when looking at the Maisp measure. The factor that would have influenced this is the fact that the development set of anchors was from an earlier benchmark [7]. Therefore, the test set of anchors could have been different in terms of content and multimodal opportunities, and in fact, could have resulted in other judgments from the crowd sourcing platform. The relevance judgment criteria could have been changed as well, what was relevant in the earlier benchmark could have been flagged non-relevant now, especially

when the anchors in this year's benchmark were created with much more multi-modality in mind than in previous years. Also, because the weights are tuned on the development set, these anchors could have influenced the used weights; while if a more random distribution of anchors was used, the weights would have been tuned to a broader set of anchors.

Another issue is the possible over-tuning of weights. There are a lot of factors that contribute to the final result set. Each parameter has its weight in determining the final ranking – some are thresholds: determining whether or not a term will be part of the query string, some are boost weights, telling Elastic Search what terms or fields are especially important. Lastly, some weights perform final re-ranking. When spending too much time to precisely tune these weights and combinations thereof to their best results, it could happen that letting these weights run on a slightly different set of data (as is the possible case for the test anchors), the weights suddenly cannot be ideal anymore. This is a problem known in machine learning as "overfitting". The problem is that it is difficult to tell and determine if the tuning process was too specific, because there are so many different factors that all have been tested on the development set. The influence of multimodal opportunities that were not used to their advantage is probably of more influence to the test scores than the possibility of over-tuning.

The fact that the other teams use the multi-modal features to their advantage in this year's benchmark is the most probable reason of the low results of this works' runs. Since the techniques used in this work focus heavily on the textual side, the visual concept side has been left aside, apart from some basic weighted combination that is used to include concepts in the query. Improvement on this part is definitely possible.

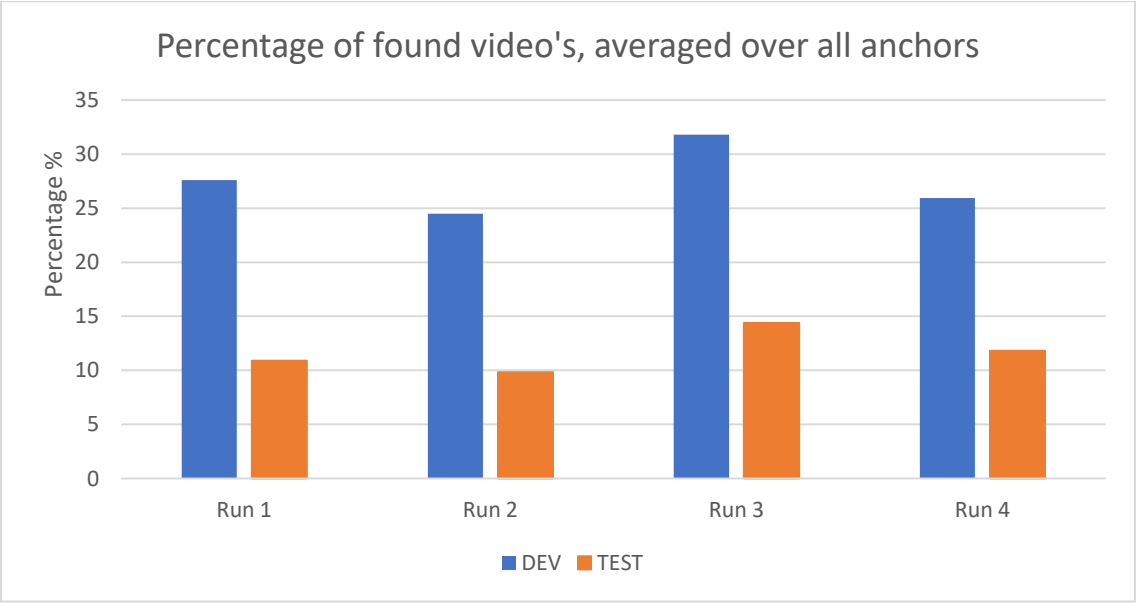
When looking at a random sample of data that the runs produced, the TF-IDF run produced sometimes additional words in the query that would not have been captured by part-of-speech sentences. This could be the reason that the results are higher for that run, since they are sometimes crucial words. An example is "domestic" in a query about "domestic abuse". Without this word, the results of the first run also contain results about animal abuse etc. The part-of-speech sentences sometimes have words along the keyword that generate noise. This is for example apparent for anchor 49, the results are for each run about golf, but for the part-of-speech runs the results are worse, they are not about learning or teaching golf, while the other two have a specific relevant result. A problem that all runs have is the fact that long transcripts generate more keywords, and therefore longer queries. Sometimes, this results in too noisy results. Another problem with this textual approach is that in the data it can be seen that the transcripts are sometimes of low quality. Since the focus in the runs lie mostly on the transcripts, this is far from ideal. In the small development set, getting a relatively large score was good enough. However, in the much larger test set, the results are lower and because there are more results, the couple of high results don't have much influence on the score anymore. In the sample of data there was also an anchor that didn't have much transcript at all, resulting in quite random results.

5.2 Limitations

One of the possible issues that could have resulted in the differences between the development results and the actual test results are the used development anchors. The development anchors were obtained from an earlier benchmark [7], focusing on what people 'say' in the videos. This could have been the reason why the development results were higher, as the techniques in this work are also text based. Also, the benchmark uses a technique called pooling to obtain the relevance results from human assessments. Pooling is a technique to asses relevance over a subset of a collection usually formed from the top K returned documents [33]. The process of pooling is also a possible reason for the overall low results of the precision @ 5 measure. Only the top 5 results of the other teams have

been assessed for relevance. Due to this limitation, the results from this study can only say something about the amount of 'same' top results. It could certainly be that the algorithms in this study have resulted in new relevant content, which has not been judged before and therefore lowering the scores because of the false not relevant marking. In order to say something about this, a small additional investigation has been done.

In the test set's evaluation file, each anchor had ground truth for between 32 and 74 distinct videos, with an average of 59 videos. In the development set, there was ground truth for as low as 7 up to 26 videos, with an average of 17 videos per anchor. In order to receive a relevance judgment, the search system should have found results in a subset of 0.5% (average 59 out of 11482 videos). The graph below shows the average percentage of videos that also occur in the evaluation file.



It can be seen that in the test scenario, the runs match only 10 to about 15 percent with the evaluation's judgments. The rest of the runs' targets do not have relevance judgments. The graph also shows the difference between the development and test set. Where the runs matched about 25% with the evaluation during development, the results drop significantly in the test scenario, but the number of evaluated items is much lower (17 vs 59) and therefore higher percentages are easier to achieve. All in all, improvements could be made to find more relevant results in line with the other participants, although the results that have been found may be relevant as well.

Another reason the results are low is the fact that this study does not make extensive use of the visual concepts or any other visual recognition techniques. Mainly, the reason was time. Since this project had to be carried out for 30 European student credits, there was only limited time. The most reasonable work had to be the development of a baseline system and some basic linking implementations, so that the system can be used for future research in another further research project with for example those visual concepts. The same reason applies to a machine learning approach, however also the lack of a large training set also makes it more difficult to implement such approaches.

The biggest limiting issue with the project was the fact that the project changed, for example the issues regarding time and due to out-of-the-box thinking regarding the indexing strategy. This could be seen when the actual project at some points deviated from the project plan and followed the decisions from meeting. Also, the plan itself was sometimes not clear, resulting in a decision being made regarding the indexing strategy and continuing on that framework instead of evaluation it

quantitatively. Now, afterwards, the realization is formed when a more extensive quantitative evaluation was held there could be a better argument. However, the subject of flexibility regarding the segmentation is difficult to evaluate quantitative. This could also be a reason together with the focus on the linking concepts to overlook the evaluation of this subpart.

5.3 Future Work

Following the limitations section, there are several recommendations that can be linked to this research to improve upon it. Firstly, this system could be used in the next benchmark in order to have the results evaluated with the other teams and generate relevance judgments for the resulting targets. This way, the results are fully comparable instead of possibly containing false negatives. Secondly, the visual modality had a small part in this study. Further use of the visual modality is possibly beneficial when looking at the results of the other teams. The system can be evaluated again with the added visual algorithms. Thirdly, to gather better data regarding the indexing and segmentation options, a study could be carried out testing the strategies in this study in more detail as well as possible other options and evaluate them by developing a quantitative evaluation. The new method using the position and offset payloads for time information (strategy 4) can then be thoroughly reviewed from an indexing perspective. Lastly, it could be really interesting to see research on machine learning techniques applied to video hyperlinking. They are already used for images, so researching the use for moving images or applying machine learning on keyframes could be an added feature for the system.

Furthermore, another topic for future work would be how to give users the targets they really want. Currently, systems apply some sort of similarity code, resulting in similar video's – but do users actually want that? During the project the question frequently came up, as the targets are found on relevancy with some of the keywords found in the anchor's transcription. Although the results are viewed and judged by crowd workers from Amazon on relevancy, it is still unclear what relevancy actually is for video hyperlinks. Currently the system does a query for the keywords in the Elastic Search index. This results in targets that also contain these keywords. Therefore, there is some link between the anchor and target, but what the link actually entails is unclear for the system. Gaining a more semantic view on this would be very interesting adaptation for this project. There could be an additional algorithm that tries to look for links that give the selected keyword more meaning, deepen the concept of the keyword, show history about the keyword or show how it's made (if the keyword is a product). Using these 'rules' gives a better fundament to the links instead of just the fact that the keyword appears in the target or is similar to the anchor because the Cosine Similarity function shows so.

5.4 Conclusion

Recall the following research questions:

- How to represent and index the multimodal data so that time-code access to segments is possible?
- What performs better for query generation from speech: using TF-IDF or Part-of-Speech?
- What performs better for re-ranking sub-segments: using weighting or cosine similarity?

Multimodal data can be indexed in Elastic Search by using a plug-in that enables the use of the position and off-set variables for time information. This way, each word has its accompanying start time indexed in the position variable. Access to this variable is possible. Segmentation is flexible this way.

From the results it can be seen that the TF-IDF and cosine similarity functions work best for the system proposed in this study. The TF-IDF statistics outperformed the baseline system and language analyzing techniques in generating a list of keywords for the query. Cosine similarity outperformed the basic weighting for re-ranking. A final remark on the runs' performance is difficult to give, as there are a lot of results that have not been checked by the crowd workers.

The lower results show possible issues with a text-only approach. Other explanations of the low results include differences between the test and development anchor set, where the development set possibly wasn't exactly representative of the test – due to the different creation process. Over-tuning the weighting also could have influenced worse results in the evaluation. It's difficult to give a final conclusion about this. From the data it can be seen that the queries contain noise. Not all captured keywords are necessary, and sometimes even give a totally other meaning to the query, which is not wanted. The difficulty lies in what to select and what not. It seems that the TF-IDF function generates the best queries (as-in: best noise, slightly less noise etc.), while the part-of-speech methods generally return more 'bad' noise. It was expected that these methods captured the important speech segments, but this did not work well as sometimes too many extra terms are captured. A reason for this could be that the transcripts are not fault-free, but the TF-IDF method also has this problem.

The several aspects of a video hyperlinking make it a project with a lot of options for future research: the indexing part of this study could have been a separate project in order to have it better evaluated and the visual modality could use more research and development.

Bibliography

[1] J. Davidson, B. Liebald, J. Liu, P. Nandy and T. van Vleet, "The YouTube Video Recommendation System," *ACM,* p. 293, 2010.

[2] M. Eskevich, G. Jones, S. Chen, R. Aly and M. Larson, "Search and Hyperlinking Task at MediaEval 2012," in *MediaEval 2012 Workshop*, Pisa, Italy, 2012.

[3] J. Oomen, W. Kraaij and A. Smeaton, "Symbiosis between the TRECVid benchmark and video libraries at the Netherlands Institute for Sound and Vision," *International Journal on Digital Libraries,* vol. 13, no. 2, pp. 91-104, 2013.

[4] Nederlands Instituut voor Beeld en Geluid, "Naar een multimediale toekomst in Beeld en Geluid," Hilversum, Netherlands, 2015.

[5] R. Ordelman, R. Aly, M. Eskevich, B. Huet and G. Jones, "Convenient Discovery of Archived Video Using Audiovisual Hyperlinking," in *SLAM'15*, Brisbane, Australia, 2015.

[6] R. Mihalcea and A. Csomai, "Wikify! Linking Documents to Encyclopedic Knowledge," in *CIKM'07*, Lisboa, Portugal, 2007.

[7] M. Eskevich, M. Larson, R. Aly, S. Sabetghadam, G. Jones, R. Ordelman and B. Huet, "Multimodal Video-to-Video Linking: Turning to the Crowd for Insight and Evaluation," in *MMM 2017*, 2017.

[8] G. Awad, J. Fiscus, D. Joy, M. Michel, A. Smeaton, W. Kraaij, G. Quénot, M. Eskevich, R. Aly, R. Ordelman, G. Jones, B. Huet and M. Larson, "TRECVID 2016: Evaluating Video Search, Video Event Detection, Localization and Hyperlinking," in *TRECVID 2016*, 2017.

[9] R. Ordelman, M. Eskevich, R. Aly, B. Huet and G. Jones, "Defining and Evaluating Video Hyperlinking for Navigating Multimedia Archives," in *WWW 2015 Companion*, Florence, Italy, 2015.

[10] M. Larson, E. Newman and G. Jones, "Overview of VideoCLEF 2009: New Perspectives on Speech-Based Multimedia Content Enrichment," *Multilingual Information Access Evaluation II, Multimedia Experiments,* pp. 354-368, 2009.

[11] B. Huurnink, C. Snoek, M. de Rijke and A. Smeulders, "Content-Based Analysis Improves Audiovisual Archive Retrieval," *IEEE Transactions on Multimedia,* vol. Vol 14, no. 4, pp. 1166-1178, 2012.

[12] B. Merialdo, P. Pidou, M. Eskevich and B. Huet, "EURECOM at TRECVID 2016: The Adhoc Video Search and Video Hyperlinking Tasks," in *TRECVID 2016*, 2016.

[13] R. Bois, V. Vukotic, R. Sicre, C. Raymond, G. Gravier and P. Sébillot, "IRISA at TRECVID2016: Crossmodality, Multimodality and Monomodality for Videohyperlinking," in *TRECVID 2016*, 2016.

[14] B. Huet, E. Baralis, P. Garza and M. Kavoosifar, "Eurecom-Polito at TRECVID 2016: Hyperlinking task," in *TRECVID 2016*, 2016 .

[15] C. Bhatt and M. Cooper, "FXPAL Experiments for TRECVID 2016: Video Hyperlinking," in *TRECVID 2016*, 2016.

[16] A. Eliëns, H. Huurdeman, M. van de Watering and W. Bhikharie, "XIMPEL Interactive Video - Between Narrative and Game Play," *GAMEON,* vol. 2008, pp. 132-136, 2008.

[17] L. Baltussen and L. Nixon, "Linking Cultural Heritage Television To The Web: A User Perspective," in *ACM TVX*, 2015.

[18] L. Baltussen and J. Oomen, "Antiques Interactive," *Proceedings of the second international ACM workshop on Personalized access to cultural heritage,* pp. 31-32, 2012.

[19] J. Liang, J. Chen, P. Huang, X. Li, L. Jiang, Z. Lan, P. Pan, H. Fan, Q. Jin, J. Sun, Y. Chen, Y. Yang and A. Hauptmann, "Informedia @ Trecvid 2016," in *TRECVID 2016*, 2016.

[20] R. Ordelman, "A Short Video Introduction to Video Hyperlinking in Dutch," 2 Februari 2016. [Online]. Available: https://videohyperlinking.com/2016/02/05/a-short-video-introduction-to-video-hyperlinking-in-dutch/. [Accessed 17 Juli 2017].

[21] M. Eskevich, M. W. and G. Jones, "New Metrics for Meaningful Evaluation of Informally Structured Speech Retrieval," *Advances in Information Retrieval. Lecture Notes in Computer Science,* 2012.

[22] D. Racca and G. Jones, "Evaluating Search and Hyperlinking: An Example of the Design, Test, Refine Cycle for Metric Development," *MediaEval 2015 Workshop,* 2015.

[23] K. McGuinness, R. Aly, F. De Jong, K. Chatfield, O. Parkhi, R. Arandjelovic, A. Zisserman, M. Douze and C. Schmid, "The AXES PRO Video Search System," in *ICMR*, Dallas, 2013.

[24] R. Baeza-Yates and G. Navarro, "Integrating Contents and Structure in Text Retrieval," *SIGMOD,* vol. 25, pp. 67-79, 1996.

[25] N. Leavitt, "Will NoSQL Databases Live Up to Their Promise?," IEEE Computer Society, 2010.

[26] D. Hiemstra and R. Baeza-Yates, "Structured Text Retrieval Models," in *Encyclopedia of Database Systems*, Berlin, Springer Verlag, 2009, pp. 2868-2871.

[27] P. Over, G. Awad, J. Fiscus, M. Michel, D. Joy, A. Smeaton, W. Kraaij, G. Quénot, R. Ordelman and R. Aly, "TRECVID 2015 - An Overview of the Goals, Tasks, Data, Evaluation Mechanisisms, and Metrics," 2016.

[28] Z. Cheng, X. Li, J. Shen and A. Hauptmann, "CMU-SMU @ TRECVID 2015: Video Hyperlinking," 2015.

[29] S. Chen, K. R. D. Curtis, L. Zhou, G. Jones and N. O'Connor, "DCU ADAPT @ TRECVid 2015: Video Hyperlinking Task," 2015.

[30] P. Galuscakova, P. Pecina, M. Krulis and J. Lokoc, "CUNI at MediaEval 2014 Search and Hyperlinking Task: Visual and Prosodic Features in Hyperlinking," 2014.

[31] R. Bois, A. Simon, R. Sicre, G. Gravier and P. Sébillot, "IRISA at TrecVid2015: Leveraging

Multimodal LDA for Video Hyperlinking," 2015.

[32] R. Bois, V. Vukotic, R. Sicre, C. Raymond, G. Gravier and P. Sébillot, "IRISA at TRECVid2016: Crossmodality, Multimodality and Monomodality for Video Hyperlinking," 2016.

[33] C. Manning, P. Raghavan and H. Schütze, Introduction to Information Retrieval, Cambridge: Cambridge University Press, 2008.

[34] D. Jurafsky and J. Martin, Speech and Language Processing, Pearson Education (US), 2008.

[35] M. Gupta and M. Bendersky, "Information Retrieval with Verbose Queries," *Foundations and Trends in Information Retrieval,* vol. 9, no. 3-4, pp. 209-354, 2015.

[36] L. Pang and C. Ngo, "VIREO @ TRECVID 2015: Video Hyperlinking (LNK)," 2015.

[37] P. Galuscakova, M. Batko, M. Krulis, J. Lokoc, D. Novak and P. Pecina, "CUNI at TRECVID 2015 Video Hyperlinking Task," 2015.

[38] E. Esen, S. Özkan and U. I., "TUZ at TRECVID 2015: Video Hyperlinking Task," 2015.

[39] S. Wang, H. Liu, Y. Xia, Y. Wang and Z. Chen, "IIPWHU @ TRECVID 2015 Video Hyperlinking," 2015.

[40] J. Barrios, R. F., J. Saavedra and D. Contreras, "ORAND at TRECVID 2015: Instance Search and Video Hyperlinking Tasks," 2015.

[41] S. Schmiedeke, P. Xu, I. Ferrané, M. Eskevich, C. Kofler, M. Larson, Y. Estève, L. Lamel, G. Jones and T. Sikora, "Blip10000: A Social Video Dataset containing SPUG Content for Tagging and Retrieval," *Proceedings of ACM Multimedia Systems Conference,* 2013.

[42] A. Krizhevsky, I. Sutskever and G. Hinton, "Imagenet Classification with Deep Convolutional Neural Networks," *Advances in Neural Information Processing Systems,* vol. 25, pp. 1106-1114, 2012.

[43] R. Aly, R. Ordelman, M. Eskevich, G. Jones and S. Chen, "Linking Inside a Video Collection - What and How to Measure," *Proceedings of the 22nd International Conference on World Wide Web,* pp. 457-460, 2013.

[44] A. Ng, "www.quora.com," 03 November 2016. [Online]. Available: https://www.quora.com/What-is-an-intuitive-explanation-of-Convolutional-Neural-Networks. [Accessed 30 September 2017].