Content Analysis of videos using Alignment Techniques

A DISSERTATION

Submitted in the partial fulfillment of the

requirements for the award of the degree of

Master of Technology

in

Computer Science and Engineering

by

SHUBHANGI JAISWAL

(15535039)

Department of Computer Science and Engineering

Indian Institute of Technology, Roorkee

Roorkee- 247667 (INDIA)

NOVEMBER, 2017

Candidate Declaration

I declare that the work presented in this dissertation with title **"Content Analysis of videos using Alignment Techniques"** towards the fulfillment of the requirement for the award of degree of **Master of Technology** in **Computer Science and Engineering** submitted in the **Department of Computer Science and Engineering, Indian Institute of Technology, Roorkee** is an authentic record of my own work carried out during the period from **July 2016** to **November 2017** under the supervision of **Dr. Manoj Misra**, Associate Professor, Department of Computer Science and Engineering, IIT Roorkee.

The content of this dissertation has not been submitted by me for the award of any other degree of this or any other institute.

DATE: ..... SIGNED: .....

PLACE: ..... (SHUBHANGI JAISWAL)

This is to certify that the statement made by the candidate is correct to the best of my knowledge and belief.

DATE: ..... SIGNED: .....

(DR. MANOJ MISRA)

Associate Professor

Dept. of CSE, IIT Roorkee

ACKNOWLEDGEMENT

First and foremost, I would like to express my utmost gratitude towards my guide Dr. MANOJ MISRA, Associate Professor, Department of Computer Science and Engineering, IIT Roorkee for his constant support and invaluable guidance throughout my dissertation research. His wisdom, knowledge and commitment to the highest standards inspired and motivated me. His encouragement and motivation have always emboldened my resolve and helped me strive higher and overcome all difficulties.

Lastly, I owe everything to the Almighty and my parents. The support which I enjoyed from my family members provided me the mental support I needed to grow as a student and a good person.

ABSTRACT

In recent years, with increase in the use of Internet the multimedia contents on it have rapidly increased. Users may want to go through a video in top down manner i.e. browsing the videos, or in bottom up manner i.e. retrieving specific information from videos. They may also want to go through the summary or through the highlights of the videos. Video data is a major multimedia data available over the web; people want interactions to be possible with the videos. This has necessitated the need to handle multimedia resources effectively. Lecture videos are the category of videos that intrigue the users to interact with the videos. This dissertation work proposes an automatic method for aligning scripts of lecture videos with captions. Alignment is needed to extract time information from captions and insert it in the scripts to create index of the videos. No alignment work has been previously done in lecture videos domain. Alignment methods proposed for other type of videos are not applicable for lecture videos because, different similarity techniques behave differently on different types of datasets. The proposed method uses transcripts of lecture videos, SRT file of captions available along with lecture videos and caption files generated from auto-caption generation feature of YouTube. The captions and scripts are then aligned using a dynamic programming technique. No such work has been previously done for lecture videos. Most important aspect of alignment is similarity measure. In the proposed work we have used three similarity measures cosine, jaccard, and dice. A comparative analysis of these measures is given in the dissertation. We also use a large lexical database of English words known as WordNet for word-to-word similarity. The experimental result shows comparison of accuracy of alignment for various similarity techniques and comparison of accuracy of alignment for captions available along with lecture videos and captions generated from YouTube's auto caption generation feature.

Contents

List of Figures

List of Tables

CHAPTER 1

INTRODUCTION

Huge amount of multimedia is today available on Internet. With growing technologies one can find videos on every domain on Internet today. Videos involve visual and hearing sense of users. They are much more interesting and informative than only written or only audible documents. So, video retrieval is an interesting topic for the researchers. Videos are used in many spheres ranging from e-learning, movies, news-broadcasts, sports etc. With the large number videos available on World Wide Web there is a need to make these videos efficient. Researchers have contributed to different types of works for video retrieval. The work includes summarization of the videos, indexing of the videos, browsing of the videos etc. Initially users were limited to the television sets, but today with the wide usage of Internet there is variety of videos available to the users. These videos can be divided into types entertainment Videos and How-To videos. For different types of videos, different interactions from users are possible. For example

Meeting Videos: It is not possible for people to attend all the meetings scheduled for them, in such cases user wants to go through the summary of the meetings. Summarization of video will allow the user to skim through the videos and understand the essence of the meetings. The aim of summarization is to include all the important points that are discussed in a meeting [1].

Movies / News Videos: Such type of videos follows a definite structure they are made by

experts. There is no uncertainty in the videos. One type of work which can be done in such videos is classification of scenes. Scenes can be classified by extracting key frames from the videos and then comparing them with a training set. The training sets are either available or we can make the machine learn by using supervised or unsupervised learning techniques. Speaker identification, audio tracks and textual image from the videos can be used for indexing the videos. Movies and news videos are scripted videos. They have a well defined structure. This scripted structure makes it easier for the researcher to come up with a generalized approach for such videos [1].

Sports Video: Users may want to watch the defining or key incidents of sports video. Sports video follows a certain type of repetitive behavior for example in the cricket video there is a repetition of the bowler bowling, batsman playing the shots, batsman running in between the wickets etc. Player playing a shot followed by a louder reaction of the crowd signifies some of the highlight of the match, so the crowd loudness matching with some of the key frame can be used for creating the highlights of the videos. Similar shots can be captured in all sports and highlights of sports videos can be generated. Also Indexing of sports video is possible; the highlights created in previous steps can be used as index so that the user can directly go to that particular position to watch the video from there. If we do not create the index of videos, the user has to move the cursor in the seek-bar to move to a particular shot. This activity is very much time consuming and also irritating when you cannot reach to a particular point where you want to switch to, this is very frequent if in case the video is of long hours [1].

Cooking Video: Researchers have proposed text based analysis of the cooking videos. The alignment of the scripts of the cooking videos which is available on the recipe website and the captions of the video can help in creating index and semantic annotation of the video. With globalization all variety of cooking videos are available on-line, in some cases we don't know what ingredients are being used in the video. Then we go through the search engines searching for the objects which are used in the videos. Sometimes we may not be able to find out the correct object which we are searching. So semantic annotators can be

added to these objects. Sometimes user may just want to know the amount of a particular ingredient being used in the recipe. In such case index of the video can be helpful to go directly to that object and see how much amount is required [1].

An analogy: A book's TOC (Table of Content) can be compared with video browsing. TOC of a book helps in browsing the contents of a book, similar technique can be used for browsing the videos. The table of content of a book can help us to know what the chapters in a book are; we can browse the entire book by going through the table of contents. Similarly a TOC of the video can be created, which can tell what the main content in the video are; it is similar to video browsing.

The index of the book can help in retrieval of contents from the book; similarly the index of a video can help in retrieval of contents from the video. Index includes the keywords in a book along with the page number. Similarly an index of the video can be created with the keywords in the videos along with the seek-bar location. We can then add some click-able areas on to the videos. So that by clicking on them we are redirected to a particular location in the video or to some other page containing some information about that object [1].

This work primarily focuses only on educational videos. There are millions of educational videos available on World Wide Web on different platforms. A few of the popular platforms are NPTEL, Khan Academy, Courseera, edX, MIT OpenCourseWare and YouTube etc. These videos have a huge number of audiences. With the large number of videos available on-line for a single topic it becomes difficult to decide which one is a good video. Students today are having vast resources but are not having the proper wisdom to choose which videos can be of use to them. This presents an opportunity to make these videos much more meaningful to the users.

This work aims at creating automatic index of the lecture videos. Users typically watch a lecture video in their entirety but sometimes they only want to go to a specific part of the video, or they want to start from a specific point in the video. At this point they need indexing to provide pinpoint access, which can also be time-saving. This can be done by using the alignment technique. In [2] authors have done work on cooking videos, where

they extract recipes and cooking videos using web crawler. They then apply caption recipe alignment algorithm on recipe and captions of videos. For comparing the similarity between sentences of recipe and captions they have used cosine similarity technique. This similarity technique may be suitable for recipe dataset, but it may not give best results in case of lecture videos dataset [3]. The dissertation's main contributions include applying different similarity measures on sentences of captions and scripts of lecture videos extracted from MIT Open Courseware, alignment of sentences from captions and script file and then comparison of performance of different similarity measures. Instead of relying only on word based similarity measures, we have also applied syntactic similarity measures which include cosine similarity, dice similarity and jaccard similarity. A comparative analysis of these techniques has been done in this dissertation. In this work, we also use a large lexical database WordNet to calculate word-to-word semantic similarity and NLTK (Natural language toolkit) to remove stopwords. We have also applied same techniques on captions generated from auto caption generation feature of YouTube. Results show that accuracy in case of captions generated by YouTube is lesser. As per our knowledge no such work has been done earlier on lecture videos.

1.1 Motivation

In the past few years there is a rapid growth in the multimedia resources available on web. Users may need to go through a video in a top down manner i.e. browsing the videos, or in bottom up manner i.e. retrieving specific information from videos. They may also want to go through the summary or through the highlights of the videos. This has necessitated the need to handle multimedia resources effectively. Video content over the web may be scripted or unscripted. Scripted videos are those videos which are having a proper structure, they are then edited afterwards and then distributed to the end users. For e.g. videos of movies and news videos are scripted videos. Whereas the unscripted videos are those which are not having any defined structure, for e.g. sports videos, meeting videos and videos of

some spontaneous events are termed as unscripted video. There is significant amount of research going on for video retrieval of both scripted as well as unscripted content. Many of the researchers use machine learning and image processing techniques such as object tracking and object recognition for video retrieval. Here accuracy completely depends upon the accuracy with which an object is recognized. The limitation of this approach is most of the videos does not have objects available as metadata. In such cases we can make use of the textual data available along with the lecture videos. Textual data is very important for video retrieval, but very few sources provide textual metadata along with the lecture videos. A huge amount of work has been done on multimedia resources but very few concentrates on the lecture videos. This gives us an opportunity to work in the field of educational videos. Our objective of this report is to gather all the previous work done on the videos of different genres and various similarity techniques used for comparison of textual documents. We also aim to find out the scope for improving the lecture videos available online and also to implement one of the approaches.

1.2 Problem statement

A video for learning purpose such as lecture videos have associated textual content. This textual content is the transcripts of the videos, and subtitle file provided in SRT format. While going through such videos a naive learner faces certain kind of problems while try to learn something new such as:

1) He may not be familiar with the terms associated in the video content.

2) He may try to search some of the keywords in the video?

3) Searching such keywords he may get wrong information from search engines.

The input to our proposed system is the transcripts of videos provided by source from where the video is taken, caption files in SRT format provided by source, caption file that is generated by speech to text conversion.

Goal of our system is to utilize this textual metadata to make the videos interactive. To do

this first objective is restructuring the SRT file according to the transcripts. Second objective is aligning the restructured caption file with the transcript file. This alignment helps in inserting time information from captions to the scripts. Third objective, of our work is using different similarity measures for calculation of similarity of documents. This results in the comparative analysis of cosine, jaccard, dice similarity measures. This analysis also states the dependency of the similarity measures on the data type. The last and final objective of our work is to, give an analysis of the importance of the textual metadata of the videos. We use captions generated by speech to text conversion, from the videos. Then we state the importance of textual metadata provided by the source of the videos, by comparing the accuracy of alignment.

1.3 Dissertation contribution

Lecture videos constitute an interesting resource for learners to learn anything they wish by visual representation. Videos involve visual and hearing sense of users. They are much more interesting and informative than only written or only audible documents. Research also states that visual learning is faster than other ways of learning. This work proposes a novel approach to create interactive lecture videos. The proposed approach comprises of two processing steps.

The first step is resource capturing. Resource capturing phase gathers the inputs which are required in the implementation of the work. The resources fetched are standard datasets.

The second step is comparison and analysis, this phase gets the input from the similarity matrix. Similarity matrix is filled using different similarity measures. Here in this work we have used cosine, jaccard, and dice similarity measure. Alignment manager is responsible for creating interactive videos. The scripts and captions of the video are passed to the alignment manager for aligning script with their subtitles, after preprocessing. This phase inserts the time information from captions to the scripts. Based on the alignment accuracy we can also compare the results of various similarity measures. We also apply the same

technique to the captions generated from speech to text conversion technique. To do this we are using auto caption generation feature of YouTube. Based on the accuracy of alignment of these captions with scripts we can predict the importance of textual metadata of the videos. A user interface can then be created which includes the functions which can be used by users for direct interaction with the videos.

Work done in this research is generalized as it can also be extended to other languages of lecture videos. The sites like COURSERA, NPTEL ,MIT provides the captions of the videos in many different languages and also the transcripts are available in different languages. If we have transcripts and captions we can directly apply the proposed method on them.

1.4 Dissertation organization

The rest of the report is organized as follows:

Chapter 2 provides the background of the different works done on alignment of captions and scripts. It also covers different similarity measures that are used for comparing textual documents.

Chapter 3 describes the proposed architecture and also the modules implemented in the work. It includes speech to text conversion module, alignment manager module, similarity measure module and comparator module etc.

Chapter 4 covers the experimental results obtained and the comparison of the proposed work with previous work.

Chapter 5 concludes with some suggestions for future research.

CHAPTER 2

BACKGROUND AND RELATED WORK

This section briefly discusses background and technologies used in this research work.

2.1 Video annotation

Amount of videos on Internet is increasing every day. Users want to search videos or specific parts of videos. They want to search videos on the basis of objects appeared in the videos, dialogues or keywords that appears in a video. Due to the increase of the videos on multimedia and changing needs of the users, there is a need to make the videos user-friendly. Researchers are focusing on establishing the relationship of user's behavior to the videos data. Video annotations can play a essential role for creating user friendly videos.

There are two ways in which the objects in the videos can be identified for annotation purpose. One is to use image processing technique along with machine learning techniques [4] [5] And another is to collect meta-data of the videos from the available textual information [6][7][8]. There are pros and cons associated with both the methods. In using image processing techniques along with machine learning techniques there is a need of high accuracy to detect and identify the object appeared in the video. Detection of objects in a video is a herculean task because of the complexities involved in the problem itself. The types of objects appeared in the videos can have a very wide range and learning of those many objects in a single system is a nearly impossible task specially when there is a requirement for high

accuracy. When textual information is considered for making meta-data of the video then there is a requirement of textual information about the video which is to be provided by the video owner or a meta-data provided by some other party. Textual information is more powerful then the object identification because textual information often describe the object of the video clearly. Textual metadata that is available with the videos includes transcripts of videos, caption files of the dialogues in the videos or text that appear in the videos. Authors in [9] have given a survey of text detection, object detection techniques for video retrieval.

2.2 Genre-specific semantic video indexing

There are different kinds of videos available on different video sharing sites. Videos can be classified into different types based on the genre of the video. Different users may watch different kinds of videos for instance for pet lover's videos having pets must be interesting. Some users may be interested in movies videos or some may be interested in the videos having an anchor. In movies videos there is no anchor, actors in movies are not focusing on the camera whereas lecture videos always have anchor who in facing and focusing on the camera. Clearly there may be various genres of the videos which are relevant for various applications.

Research has been done for the classical problem of learning these genres to system. This Problem in itself is difficult to resolve but application oriented solutions are easy to deliver for such problems.

In [5] one approach for detecting genre based videos is proposed, the approach is a two step framework which first learn the labeled videos data which is labeled at both video label as well as shot level. At the video level the data is been labeled for the genre of the video and at shot label it is labeled as the semantic concept of the shot. Classification is applied for reducing the entire data to a relatively smaller dataset and then genre specific learning model is applied for that smaller dataset.

2.3 Text based alignment

In [7] authors have presented a framework for aligning and indexing movie with their script. The framework uses structural units such as shots, actions, scenes and dialogs from the movie and aligning them based on the longest common subsequence. Researchers have tried to provide a formal way to align caption files of movies with their scripts and provided learning mode files for the objects, scenes and actors. In [7] authors have constructed grammar rules for parsing the continuity scripts of movie. By constructing grammar the script can be analyzed as a regular expression. Grammar can also be used for automatically transcribing the entire script into XML tree.

In [8] authors have presented an approach for the speaker or character recognition by making use of screenplay. High level information is difficult to extract from the audiovisual components, so a technique is required which can provide better feature extraction, processing and analysis of the screenplay. This method make use of screenplay by parsing screenplay and aligning it with time-stamped captions of the movie videos, and then recognizing audio source by breaking the audio and mapping it with the screenplay.

Screenplay provides all the information of a movie, which includes dialogues of the characters, the information about the sets where the movie is shot. It is difficult to use the screenplay for the content-based analysis due to the following challenges: 1) since formatting of the screenplay is not same for all the movies it becomes difficult to parse the screenplay. 2) The time-stamp information is not present in the screenplay. 3) The scenes and dialogues can be modified, deleted, added, and removed from the movie. These dynamic changes may not be recorded in the screenplay. Authors in [10] have used thesaurus along with dynamic programming techniques on movie scripts and captions which has considerably improved the results. Authors in [11] have further improved the above stated algorithms by giving more weight to the matches and also considering scene reordering while alignment. For comparing the similarity between two documents they have considered entire document as one sentence and applied word based similarity. Authors in [8] have also created a word

based similarity matrix where similarity matrix is filled with ones or zeros depending on whether each word is same or not. They have used grammar rules for parsing the caption file. In [2] the technique is applied on cooking videos instead of movie videos. Here instead of creating a word based similarity matrix a sentence based similarity matrix is created. The values of this matrix are filled by using cosine similarity measure and DP algorithm is used to find optimal path in the matrix. In [12] authors have used semantic annotations on cooking videos. They have extended the work proposed in [2]. Authors in [13] have also extracted text of lecture videos, but instead of depending on external textual files authors have extracted texts from video scenes by detection of texts.

2.4 Sentence similarity measures

While aligning two sentences we need to compare whether the two sentences are similar or not. There are various similarity measures that can be used for comparing two sentences. Similarity measure is the distance between various data points. The similarity between sentences becomes an important aspect while comparing two sentences for any application. Authors in [3] have used different syntactic similarity measures to compute similarity between sentences and given a comparative analysis between cosine similarity, jaccard similarity and dice similarity. They have concluded that jaccard and dice performs better as compared to cosine similarity when sentences are of smaller units and also when documents are decomposed into sentences. Authors in [14] have given a comparative analysis of following measures. 1) Sentence semantic similarity measure, 2) Word order similarity, 3) Combined semantic and syntactic measure. Different similarity measures have different pros and cons on different type of datasets. Determining the similarity between sentences have large impact in many text applications [15] [16].

2.5 Observations and research possibility

Some of the important observations that have been made from this literature review are

1) Although lot of work has been done on the movie videos for caption script alignment, but the effort for possibilities of making lecture videos interactive is not properly explored. In the next section a system is proposed for making the lecture videos interactive.

2) The alignment accuracy for aligning script with captions can be improved by using different similarity measures.

3) Better processing approaches can be used to improve the accuracy of alignment.

4) All the videos do not provide the textual metadata. Textual metadata is very important to make videos interactive. We can show some analysis which can prove that authentic textual metadata provided by the source that owns the video, is important to make videos interactive.

CHAPTER 3

SYSTEM ARCHITECTURE

We propose a system for automatic generation of interactive lecture videos. The chapter provides the modular structure of entire architecture of the system. The chapter elaborates role of each of the module in our architecture.

The architectural overview of the system is shown in Figure 3.1. The architecture consists of two phases one is resource capturing and the other is comparison and analysis phase. The resource capturing phase captures videos and textual content of lecture videos from the websites using a web crawler. Different educational websites have different types of content and also in different formats, so customized crawlers will be needed for different websites. It also consists of preprocessing and restructuring module and speech to text conversion module. The restructuring module restructures the SRT file according to the scripts of the video so that they are in comparable formats. The preprocessing module removes stopwords and uses wordnet to find synonyms of the words. We use natural language toolkit for preprocessing. Preprocessing of scripts and caption files results in better accuracy of alignment. Speech to text conversion module uses auto caption generation feature of YouTube to generate captions from the videos. This module states the importance of textual metadata along with the videos. Second phase consist of Similarity measure module, alignment manager module, and comparator module. Similarity measure takes into consideration all the similarity index for comparing two sentences. The alignment manager

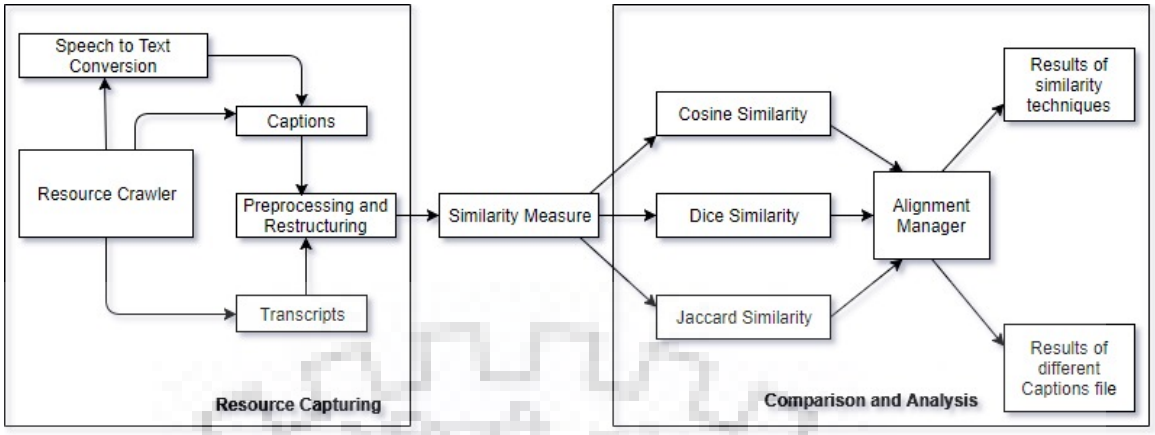


Figure 3.1: *System Architecture*

align the captions file and the scripts of the videos. Scripts include all the information of a video except the time information. The alignment manager extracts time information from the captions and inserts it into the processed scripts. The comparator module compares the results of various similarity indexes for different alignment techniques. The comparator module also compares the results of captions that are directly taken from the source of the videos and results of captions that are taken from speech to text conversion procedure.

3.1 Modules in our system

The system consists of following modules as listed below:-

1) Resource crawler

2) Speech to text conversion

3) Preprocessing and restructuring

4) Similarity measure

5) Alignment manager

6) Comparator

Description of each of the above modules is provided in the subsections 3.2 to 3.7.

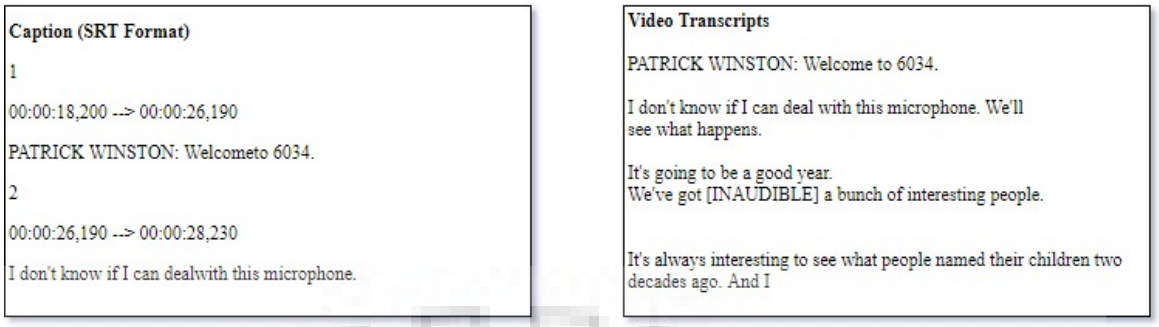


Figure 3.2: *An example of caption in SRT format and video transcripts*

3.2 Resource crawler

At the resource crawling phase a web crawler extract the textual metadata from different educational websites. This textual metadata consists of scripts of lecture videos and subtitle files of these lecture videos which are generally available in SRT format. Different lecture video websites have different predefined formats so different types of crawler will be needed in order to support the extraction of video and text from different websites. Since we are only using textual metadata for our work, we will focus only on the textual metadata. Figure 3.2 shows scripts and captions of a lecture videos extracted from one of the lectures of MIT Open Courseware.

3.3 Speech to text conversion

There is significant amount of research in the field of video retrieval. Many of the researchers use machine learning and image recognition techniques such as object tracking and object recognition for video retrieval. Here accuracy completely depends upon the accuracy with which an object is recognized. The limitation of this approach is most of the videos does not have objects available as metadata. In such cases we can make use of the textual data available along with the lecture videos. Textual data is very important for video retrieval, but very few sources provide textual metadata along with the lecture videos. In case complete

textual data is not provided by the owners of the videos we can use speech to text conversion technique to generate textual data. Here in this work we are using YouTube's auto caption generation feature to generate the captions file. We can use this metadata for video retrieval but the accuracy of this data as compared to the data provided by the video owners is much less. We use both the captions files, one which we get from the actual video owners and one which is generated by speech to text conversion technique. We align both the caption files with scripts of the videos and state the difference in accuracy of alignment. The results of this are stated in section 4.3. This analysis states the importance of authentic and accurate textual metadata provided by video owners.

3.4 Preprocessing and restructuring

Captions and scripts file extracted from the resource crawling phase cannot be aligned directly because they are in different formats. In order to get better results both the files should be in the same format. For this, we are restructuring the captions file. Captions in the SRT format do not represent a sentence. The SRT format consists of the following four elements. 1) A number indicating sequence of subtitle, 2) The time point at which subtitle appears on the screen, 3) The subtitle itself for that time period, 4) A blank line indicating the start of a new subtitle. The captions are created according to the time gaps taken by the person who is speaking. By doing the restructuring of the captions the sentences are constructed out of different elements of the SRT file. Full stop '.' separates two sentences so we make use of it. Figure 3.3 shows an example of restructuring of SRT format captions file. Preprocessing of the captions and the textual content also needs to be done before the alignment. While preprocessing the sentences we remove all the non-semantic words from the sentence which are not giving context to the sentence. For example words like a, an, the, to, has, have etc. These words are mainly helping verbs, conjunctions and prepositions. They are called stopwords. By removing these words we remove the chance of unnecessary matching of words of two sentences. The match quality improves by removing non-semantic

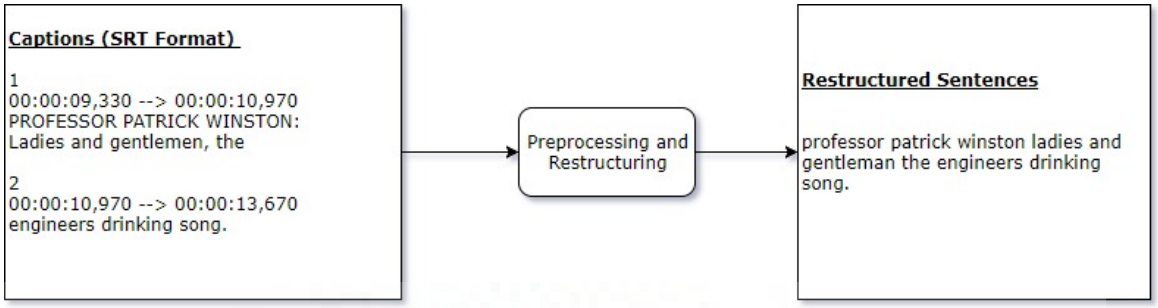


Figure 3.3: *An example showing restructuring and preprocessing of SRT format captions*

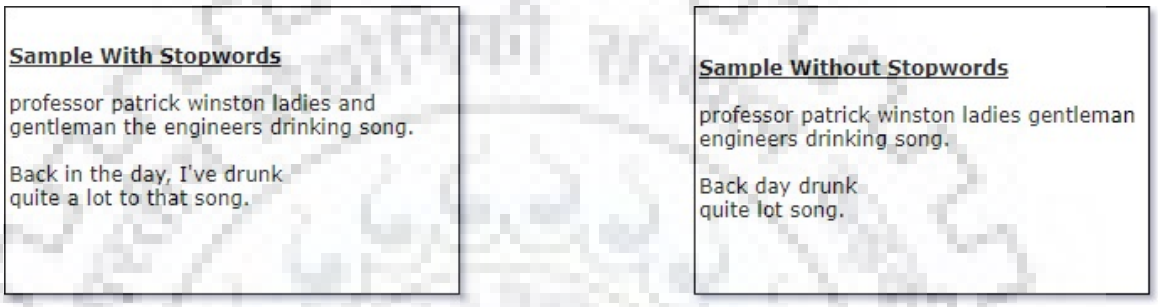


Figure 3.4: *An example showing sentences with and without STOPWORDS*

words. For this we use natural language toolkit. Preprocessing also includes the changing of numeric values to the words for example 55 in the text will not match appropriately but fifty-five is more useful for the aligning purpose. Further for preprocessing, we also find words which have similar meanings. For this we use large lexical database of English words known as WordNet for finding word-to-word similarity [17]. WordNet is a large lexical database of English, nouns, verbs, adjectives and adverbs grouped into set of sets. WordNet structure makes it a useful tool for computing linguistics and NLP. WordNet is similar to a thesaurus. It groups words based on their meanings. Figure 3.4 shows an example of preprocessing of the files where stopwords are removed and wordnet is used to find the synonyms.

3.5 Similarity Measure

When we are aligning the captions file with the script file of lecture videos, we compare each sentence in caption file with each sentence in script file. Two sentences are said to be similar

if the similarity index is high for them. According to our observations different similarity measures behave differently on different datasets. After doing a study of different similarity measures, we have chosen three standard similarity measures to compare the sentences of scripts and captions. The similarity measures which we are using are; cosine similarity, dice similarity and jaccard similarity measure. We find that in some cases cosine give better results but sometimes jaccard and dice performs better than cosine. Below are the equations which are used for calculation of cosine, jaccard and dice similarity.

1) Cosine Similarity Measure [18]

$$S_{A,B} = \frac{\|Words_A \cap Words_B\|}{\sqrt{\|Words_A\| \|Words_B\|}} \tag{3.1}$$

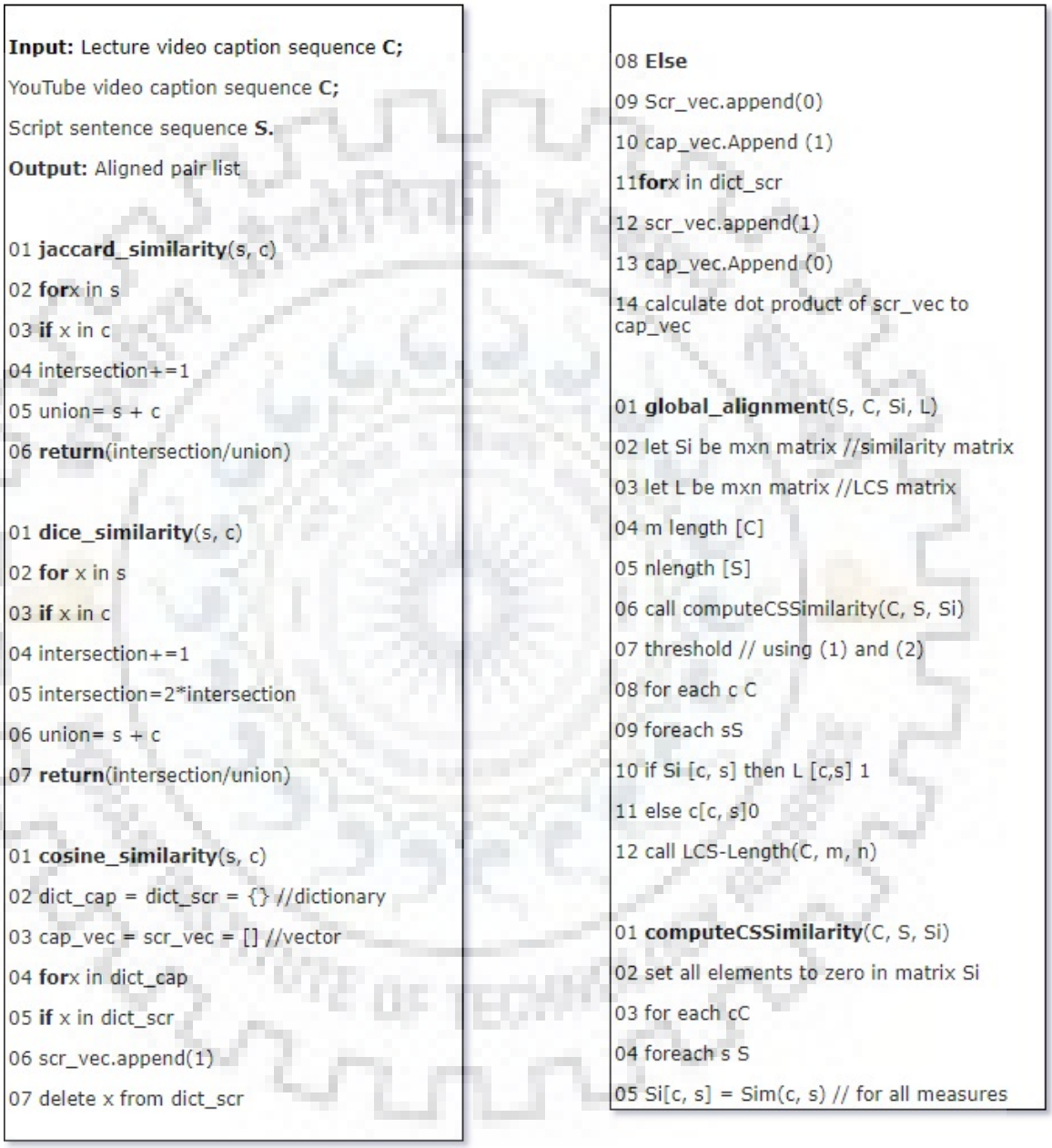
2) Dice Similarity Measure [18]

$$S_{A,B} = \frac{2\|Words_A \cap Words_B\|}{\|Words_A\| + \|Words_B\|} \tag{3.2}$$

3) Jaccard Similarity Measure [18]

$$S_{A,B} = \frac{\|Words_A \cap Words_B\|}{\|Words_A \cup Words_B\|} \tag{3.3}$$

Section 4.3 shows the results of cosine, jaccard and dice similarity measure when applied to the captions and scripts file.



```
Input: Lecture video caption sequence C;
YouTube video caption sequence C;
Script sentence sequence S.
Output: Aligned pair list

01 jaccard_similarity(s, c)
02 forx in s
03 if x in c
04 intersection+=1
05 union= s + c
06 return(intersection/union)

01 dice_similarity(s, c)
02 for x in s
03 if x in c
04 intersection+=1
05 intersection=2*intersection
06 union= s + c
07 return(intersection/union)

01 cosine_similarity(s, c)
02 dict_cap = dict_scr = {} //dictionary
03 cap_vec = scr_vec = [] //vector
04 forx in dict_cap
05 if x in dict_scr
06 scr_vec.append(1)
07 delete x from dict_scr
08 Else
09 Scr_vec.append(0)
10 cap_vec.Append (1)
11forx in dict_scr
12 scr_vec.append(1)
13 cap_vec.Append (0)
14 calculate dot product of scr_vec to
cap_vec

01 global_alignment(S, C, Si, L)
02 let Si be mxn matrix //similarity matrix
03 let L be mxn matrix //LCS matrix
04 m length [C]
05 nlength [S]
06 call computeCSSimilarity(C, S, Si)
07 threshold // using (1) and (2)
08 for each c C
09 foreach sS
10 if Si [c, s] then L [c,s] 1
11 else c[c, s]0
12 call LCS-Length(C, m, n)

01 computeCSSimilarity(C, S, Si)
02 set all elements to zero in matrix Si
03 for each cC
04 foreach s S
05 Si[c, s] = Sim(c, s) // for all measures
```

Figure 3.5: *A caption-Script alignment algorithm*

3.6 Alignment manager

Task of the alignment manager is to find the subsequences which are similar. Therefore the alignment manager finds the longest common subsequence (LCS) between the preprocessed captions and scripts of lecture videos. Researchers in [8] [7] [11] [6] applied dynamic programming (DP) approach for the identification of LCS. Using DP they successfully found the optimal path between scripts and the captions of movies. Our proposed alignment algorithm is based on the alignment of the captions and the script of the lecture videos. While instructor explains every step of the lecture videos with some examples, the scripts are written form of entire lecture videos. The sequence of the script sentences may be different from the sequence of the captions of the lecture videos. Also sometimes the caption files have more content than the scripts due to some more descriptions given by the instructor. It does not matter to the user of the lecture videos to have some differences in the captions and the scripts but it causes problem when machine handles the sentences. A similarity matrix is filled based on the similarity between the captions and the script contents. Then dynamic programming algorithm is applied to find an optimal path in the matrix. The alignment algorithm uses sentences of preprocessed captions and scripts as input and produces aligned pair list. A similarity matrix is constructed by using sentences of both inputs. Similarity value is then used to fill every cell of the similarity matrix. To measure similarity value between two sentences we use, cosine similarity, dice similarity and jaccard similarity measure. To measure the similarity index the sentences are tokenized into words and then the words are transformed into vectors. Similarity values are calculated using vector space model and then filled in the matrix. Our similarity matrix has similarity values between zero and one. Consequently, we need to decide the threshold for the valid similarity value. The threshold value for similarity is decided by the equations given below.

$$threshold = mean(similarity\_value) + \alpha * \sigma(similarity\_value) \tag{3.4}$$

$$threshold = min(max(similarity\_value_i)) - \alpha * \sigma(similarity\_value) \tag{3.5}$$

Here in (3.4) we take the average value of all similarity values and σ is the standard deviation to the mean value calculated and α is a parameter. In (3.5) we take minimum of all the maximum values of similarity index in each row of the similarity matrix. After the completion of alignment process, the time-stamp information is added to the corresponding sentences of the script. Figure 3.5 represents caption-script alignment algorithm which is used for alignment.

3.7 Comparator

In our proposed work we have used different similarity measures while aligning the captions and scripts file. We have also used captions from two different sources. This module states the better similarity index and also the importance of text material provided by the source of the lecture videos. To calculate accuracy we have used (4.1). To measure the accuracy of each of the similarity measure we have used the same equation. Based on the results we have deduced which of the similarity measure suits for what kind of data type. For different caption files also we have used (4.1) to calculate the accuracy. The difference in accuracy of both the caption files states the importance of textual metadata provided with the video files.

CHAPTER 4

EXPERIMENTS AND DISCUSSION

The proposed approach of indexing requires lecture videos, transcripts of the lecture videos provided by source, and caption files of the lecture videos provided by source, and captions generated by auto caption generation feature of YouTube for these lecture videos. Since the SRT file elements are not directly comparable to the scripts, the restructuring of the file is required. The restructuring unit then restructures the SRT elements to create sentences. We also remove stopwords from the scripts and captions file. The WordNet database is also used to preprocess the scripts and captions file. Using WordNet database we find synonyms for the words. Removing stopwords and adding synonyms improves the accuracy of similarity measure of two sentences.

This chapter presents the experimental results for the proposed work.

4.1 The Dataset and Experimental Environment

In our work we have used lecture videos from MIT OpenCourseWare. MIT OpenCourseWare provides transcripts of the lecture videos and the captions of these lecture videos in SRT format. For removing the stopwords we have used NLTK. The Natural Language Toolkit, or more commonly known as NLTK, is a suite of libraries and programs for symbolic and statistical natural language processing for English written in the Python programming

Table 4.1: *A Summary of our experimental data set*

Sample	**Source SRT Elements**	**Source SRT Elements after Refinement**	**YouTube SRT Elements**	**YouTube SRT Elements after Refinement**
1	864	378	928	381
2	425	141	837	140
3	878	511	902	502
4	858	542	901	545
5	845	725	894	712

language. It includes libraries through which stopwords can be imported and logic can be used to remove it from our dataset depending upon the type of dataset. To find synonyms of the words we have used one of the libraries of NLTK. The library contains synonyms corresponding to a word, logic can be written depending upon the dataset and synonyms of the words can be extracted. After preprocessing different similarity measures are applied these are cosine, jaccard and dice on both the captions file as mentioned above. An analysis is done on the results to state the better quality of captions and also the better similarity measure for different dataset.

Data resources that are used in this approach are summarized in Table 4.1.

'Source SRT Elements' here refers to the number of elements in the caption file received from source/owner of the lecture videos. 'Source SRT Elements after Refinement' refers to the number of SRT elements after processing through the restructuring module. 'YouTube SRT Elements' refers to the number of elements in the caption file extracted from auto caption generation feature of YouTube. 'YouTube SRT Elements after Refinement' refers to the number of SRT elements after processing through the restructuring module.

4.2 Performance Metrics

After the restructuring of the SRT elements, a similarity matrix and a LCS matrix is created. In these matrix rows represent the captions sentences and columns represent the script sentences. The similarity between two sentences is created each of the similarity index. Similarity matrix is filled based on the values calculated from similarity measure. The values in LCS matrix are filled based on threshold as in (3.4) and (3.5). To calculate the accuracy of aligned sentences following equation is used.

$$Accuracy = \frac{no.\ of\ matching\ sentences}{max(Sentences\ in\ caption\ file, Sentences\ in\ script\ file)} \tag{4.1}$$

Our algorithm shows higher accuracy as compared to the existing approaches. The proposed method provides an average increase of alignment accuracy in comparison to [2].

4.3 Experimental Results

Our system for interactive lecture videos is proposed to generate interactive lecture videos and to support the users to interact with the videos while watching the videos. For generating interactive lecture videos, the modules discussed in section 3.2 to section 3.7 are applied. Some experiments are performed for the evaluation of the proposed approach. For evaluation, we measure the accuracy of caption-script alignment for each of the similarity measure cosine, jaccard, dice. To calculate the accuracy we use (4.1). We have used captions file directly from the source and one caption file which is generated by auto caption feature generation of YouTube. Table 4.2 represents the accuracy when (3.4) is used to calculate threshold and caption file and script file is not processed using NLTK. The caption file here is one that is provided by the resource owners.

Table 4.2: *Results of files(original captions) with stopwords and mean as threshold*

Alpha	Sample 1			Sample 2			Sample 3		
	Cosine	Dice	Jaccard	Cosine	Dice	Jaccard	Cosine	Dice	Jaccard
0.1	94.7	94.17	94.17	96.99	96.99	96.99	96.46	96.07	95.87
0.2	94.44	94.17	93.65	96.99	96.99	96.99	96.46	95.87	95.87
0.3	94.44	93.65	92.85	96.99	96.99	95.48	96.07	95.87	95.67
0.4	94.44	93.38	92.85	96.99	96.99	95.48	96.07	95.87	95.28
0.5	93.91	92.59	92.85	96.99	96.99	95.48	95.67	95.28	95.08
0.6	93.91	92.32	92.32	96.99	96.99	94.73	95.48	95.28	95.08
0.7	93.65	92.06	91.79	95.48	96.24	94.73	95.48	94.89	94.89
0.8	92.85	91.79	91.53	95.48	96.24	94.73	95.48	94.89	94.89
0.9	92.85	91.26	90.47	95.48	94.73	94.73	95.48	94.89	94.69
1.0	92.59	91.26	90.47	95.48	94.73	94.73	95.48	94.89	94.49

Table 4.3 represents the accuracy of caption-script alignment when (3.4) is used to calculate threshold and caption file and script file is processed using NLTK. In this case we have removed stopwords and also used WordNet to find the synonyms of the words. The caption file here is one that is provided by the resource owners.

Table 4.4 represents the accuracy when (3.5) is used to calculate threshold and caption file and script file is not processed using NLTK. The caption file here is one that is provided by the resource owners. In min_of_max strategy we try to find at-least one match for the sentences. Table 4.5 represents the accuracy of caption-script alignment when (3.5) is used to calculate threshold and caption file and script file is processed using NLTK. In this case we have removed stopwords and also used WordNet to find the synonyms of the words. The caption file here is one that is provided by the resource owners.

To give a comparative analysis of captions obtained from the source of the video providers and captions extracted from speech to text conversion technique, we apply the same techniques on the caption file extracted from auto caption generation feature of YouTube.

Table 4.6 represents the accuracy when (3.4) is used to calculate threshold and caption file and script file is not processed using NLTK. The caption file here is one that is extracted

Table 4.3: *Results of files(original captions) without stopwords and mean as threshold*

Alpha	Sample 1			Sample 2			Sample 3		
	Cosine	Dice	Jaccard	Cosine	Dice	Jaccard	Cosine	Dice	Jaccard
0.1	94.17	94.17	94.17	97.74	97.74	97.74	95.28	95.28	95.28
0.2	94.17	94.17	94.17	97.74	97.74	96.99	95.28	95.28	95.28
0.3	94.17	94.17	94.17	97.74	97.74	96.99	95.28	95.28	95.28
0.4	94.17	94.17	94.17	96.99	96.99	96.99	95.28	95.28	95.28
0.5	94.17	94.17	94.17	96.99	96.99	96.24	95.28	95.28	95.28
0.6	94.17	94.17	94.17	96.99	96.99	95.48	95.28	95.28	95.08
0.7	94.17	94.17	94.17	96.99	96.99	95.48	95.28	95.28	95.08
0.8	94.17	94.17	94.17	96.99	96.99	95.48	95.08	95.28	94.89
0.9	94.17	94.17	93.65	96.99	95.48	95.48	95.48	95.08	94.69
1.0	94.17	94.17	93.38	96.24	95.48	95.48	92.53	95.08	94.49

from auto caption generation feature of YouTube.

Table 4.7 represents the accuracy of caption-script alignment when (3.4) is used to calculate threshold and caption file and script file is processed using NLTK. In this case we have removed stopwords and also used WordNet to find the synonyms of the words. The caption file here is one that is extracted from auto caption generation feature of YouTube.

Table 4.8 represents the accuracy when (3.5) is used to calculate threshold and caption file and script file is not processed using NLTK. The caption file here is one that is extracted from auto caption generation feature of YouTube.

Table 4.9 represents the accuracy of caption-script alignment when (3.5) is used to calculate threshold and caption file and script file is processed using NLTK. In this case we have removed stopwords and also used WordNet to find the synonyms of the words. The caption file here is one that is extracted from auto caption generation feature of YouTube.

In (3.4) and (3.5) alpha is a parameter. Table 4.2 to Table 4.9 shows results for different values of alpha. We have chosen 0.1 as the optimum value of alpha. According to our analysis the alpha value is dependent on the type of method used for threshold. We want to increase the probability of accuracy of matches. We need to ascertain that there are maximal

Table 4.4: *Results of files(original captions) with stopwords and min of max as threshold*

Alpha	Sample 1			Sample 2			Sample 3		
	Cosine	Dice	Jaccard	Cosine	Dice	Jaccard	Cosine	Dice	Jaccard
0.1	93.91	95.23	95.50	95.48	96.24	95.48	93.51	94.89	94.89
0.2	93.91	95.23	94.97	95.48	96.24	95.48	93.12	94.69	94.69
0.3	93.38	94.97	94.70	95.48	94.73	94.73	92.92	94.49	94.49
0.4	92.85	94.70	94.44	95.48	94.73	94.73	92.53	94.30	94.30
0.5	92.85	94.44	94.17	95.48	94.73	94.73	92.14	94.30	94.30
0.6	92.32	93.91	94.17	95.48	94.73	94.73	91.94	94.10	93.71
0.7	92.32	93.65	93.65	94.73	94.73	94.73	91.94	93.12	93.32
0.8	91.79	93.38	92.85	94.73	94.73	93.23	90.96	92.92	93.12
0.9	91.53	93.12	92.85	93.23	94.73	93.23	90.56	92.73	93.12
1.0	91.26	92.32	92.32	93.23	94.73	93.23	90.56	92.73	92.14

Table 4.5: *Results of files(original captions) without stopwords and min of max as threshold*

Alpha	Sample 1			Sample 2			Sample 3		
	Cosine	Dice	Jaccard	Cosine	Dice	Jaccard	Cosine	Dice	Jaccard
0.1	92.59	92.06	93.38	95.48	95.48	95.48	92.33	92.33	91.15
0.2	92.32	92.06	93.12	95.48	95.48	95.48	92.14	92.14	91.15
0.3	92.06	92.06	92.59	95.48	95.48	95.48	92.14	91.94	90.76
0.4	92.06	92.06	92.32	95.48	95.48	95.48	92.14	91.74	90.76
0.5	92.06	92.06	92.32	95.48	95.48	95.48	92.14	91.74	90.56
0.6	92.06	91.79	89.94	95.48	95.48	95.48	91.94	91.74	90.56
0.7	91.79	91.79	89.94	95.48	95.48	95.48	91.75	91.74	90.56
0.8	91.79	91.00	89.41	95.48	95.48	95.48	91.55	91.74	89.39
0.9	91.79	90.74	89.41	95.48	95.48	95.48	91.35	91.74	89.39
1.0	91.79	90.74	89.41	95.48	95.48	95.48	91.35	91.35	89.39

Table 4.6: *Results of files(STT captions) with stopwords and mean as threshold*

Alpha	Sample 1			Sample 2			Sample 3		
	Cosine	Dice	Jaccard	Cosine	Dice	Jaccard	Cosine	Dice	Jaccard
0.1	88.71	88.71	88.18	82.57	80.30	79.54	86.40	86.00	85.39
0.2	88.18	88.45	86.87	81.81	80.30	78.03	86.00	85.20	85.00
0.3	87.13	87.40	86.87	81.06	80.30	77.27	85.20	84.80	84.39
0.4	86.61	86.61	86.08	80.30	79.54	75.75	84.80	84.39	84.39
0.5	86.08	86.61	85.30	79.54	78.30	75.00	84.60	84.20	84.20
0.6	86.82	85.82	84.25	78.03	77.27	75.00	84.39	84.20	83.00
0.7	85.56	85.03	83.72	77.27	76.51	74.24	84.00	83.20	82.00
0.8	85.30	83.72	83.72	76.51	75.75	74.24	83.20	83.20	81.39
0.9	85.03	83.72	82.93	75.75	75.00	72.72	82.39	82.00	80.80
1.0	84.77	83.46	82.67	74.24	75.00	71.21	81.60	81.80	80.20

Table 4.7: *Results of files(STT captions) without stopwords and mean as threshold*

Alpha	Sample 1			Sample 2			Sample 3		
	Cosine	Dice	Jaccard	Cosine	Dice	Jaccard	Cosine	Dice	Jaccard
0.1	85.30	85.30	85.30	84.09	84.09	82.57	81.60	81.60	81.60
0.2	85.30	85.30	85.30	84.09	83.33	81.81	81.60	81.60	81.60
0.3	85.30	85.30	85.30	83.33	82.57	80.30	81.60	81.60	81.60
0.4	85.30	85.30	85.30	81.81	81.81	79.54	81.60	81.60	81.60
0.5	85.30	85.30	85.30	79.54	80.30	78.78	81.60	81.60	81.60
0.6	85.30	85.30	85.30	78.78	78.78	78.78	81.60	81.60	81.60
0.7	85.30	85.30	85.30	78.78	78.78	78.78	81.60	81.60	81.60
0.8	85.30	85.30	85.30	78.78	78.78	78.78	81.60	81.60	81.60
0.9	85.30	85.30	85.03	78.78	78.03	77.27	81.60	81.60	81.60
1.0	85.30	85.30	85.03	78.78	78.03	76.51	81.60	81.60	81.60

Table 4.8: *Results of files(STT captions) with stopwords and min of max as threshold*

Alpha	Sample 1			Sample 2			Sample 3		
	Cosine	Dice	Jaccard	Cosine	Dice	Jaccard	Cosine	Dice	Jaccard
0.1	85.56	88.97	88.18	90.90	92.42	91.66	82.00	86.00	86.60
0.2	85.30	88.71	87.13	90.15	91.66	88.63	81.38	86.00	86.00
0.3	85.03	88.45	86.87	89.39	90.15	87.87	81.00	85.20	85.20
0.4	85.03	87.40	86.08	87.87	89.39	86.36	80.60	84.80	84.60
0.5	83.98	86.61	85.30	87.12	87.12	84.09	79.80	84.39	84.39
0.6	83.72	86.61	84.25	86.36	86.36	83.33	79.60	84.20	84.20
0.7	83.46	85.82	83.98	85.60	86.36	81.06	79.40	84.20	84.20
0.8	82.93	85.03	83.72	82.33	84.84	81.06	78.80	83.20	82.80
0.9	82.67	84.25	83.20	82.57	81.81	79.54	78.80	82.00	81.80
1.0	82.15	83.72	82.67	82.57	81.81	78.78	78.60	82.00	81.39

Table 4.9: *Results of files(STT captions) without stopwords and min of max as threshold*

Alpha	Sample 1			Sample 2			Sample 3		
	Cosine	Dice	Jaccard	Cosine	Dice	Jaccard	Cosine	Dice	Jaccard
0.1	84.77	84.51	82.15	84.09	84.09	84.09	81.60	81.60	81.60
0.2	84.77	84.51	81.62	84.09	84.09	84.09	81.60	81.60	81.60
0.3	84.77	84.51	80.83	84.09	84.09	84.09	81.60	81.60	81.60
0.4	84.77	84.51	80.83	84.09	84.09	84.09	81.60	81.60	81.60
0.5	84.77	84.51	80.83	84.09	84.09	83.33	81.60	81.60	81.60
0.6	84.51	84.51	80.83	84.09	84.09	82.57	81.60	81.60	81.60
0.7	84.51	83.98	80.83	84.09	84.09	81.06	81.60	81.60	81.60
0.8	84.51	83.46	80.83	84.09	84.09	80.30	81.60	81.60	81.60
0.9	84.51	83.46	80.05	82.57	82.57	79.54	81.60	81.60	81.60
1.0	84.25	82.15	80.05	81.06	81.81	78.78	81.60	81.60	81.60

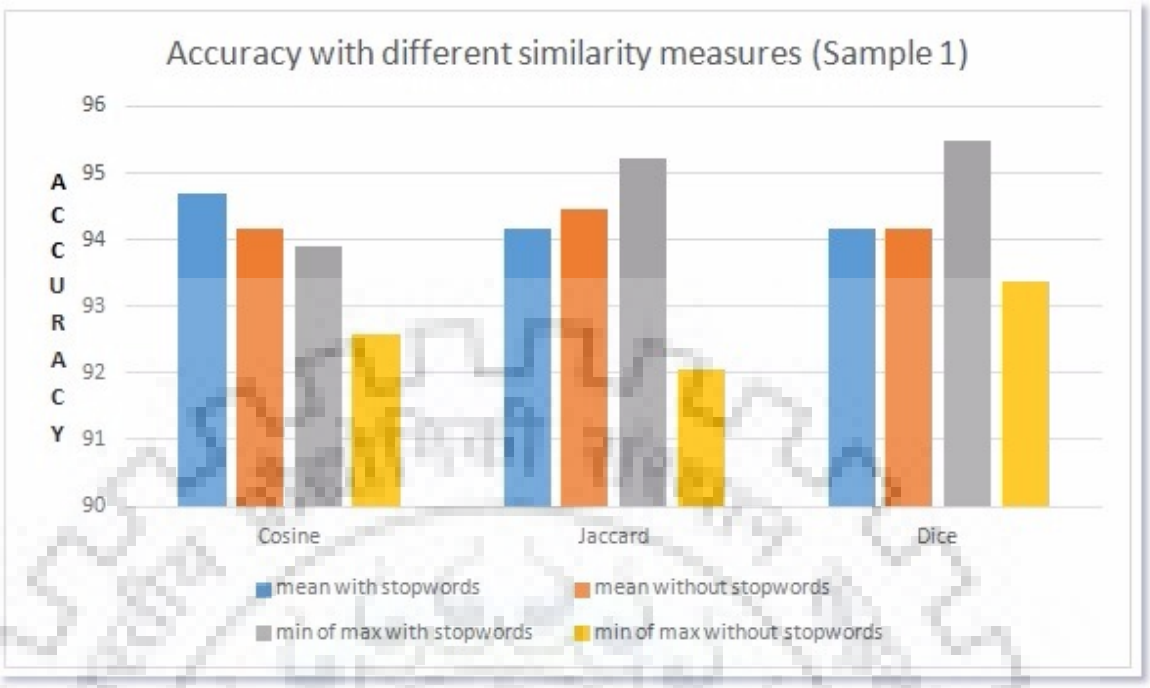


Figure 4.1: *Results for Sample 1 (original captions)*

matches for our sentences. In case of (3.4), here we have used mean as the measure for threshold. All values in the similarity matrix which have values greater than mean will get a value '1' in LCS matrix and others are marked as '0' in LCS matrix. The accuracy will depend on number of '1's in the LCS matrix. Minimum value of alpha will give maximum '1's. In case of (3.5) where we have used min of max, here we want to increase the number of matches for any sentence, but we also want that the two sentences which are matched are exact. To make sure maximum matching sentences are aligned here in equation (3.5) also 0.1 value of alpha is the optimal. We have chosen 0.1 to be optimal value, and shown a graphical representation of accuracy. We have plotted accuracy for three samples as shown in Figure 4.1 to Figure 4.6, where we have taken values from each of the tables from Table 4.2 to Table 4.9.

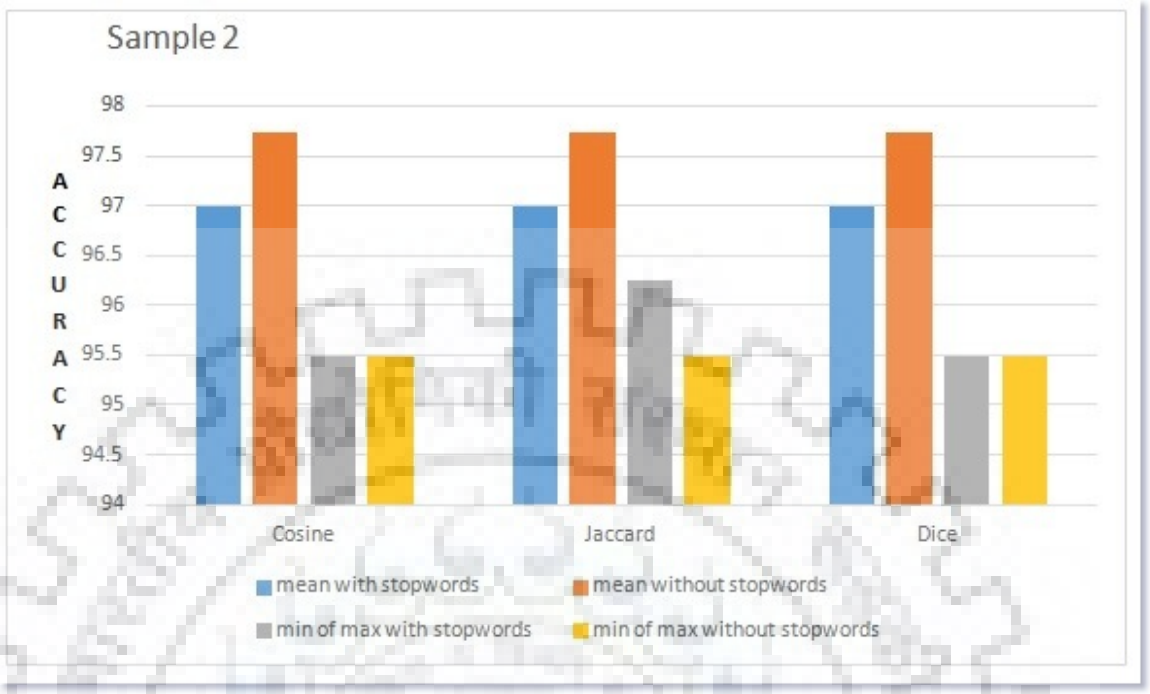


Figure 4.2: *Results for Sample 2 (original captions)*

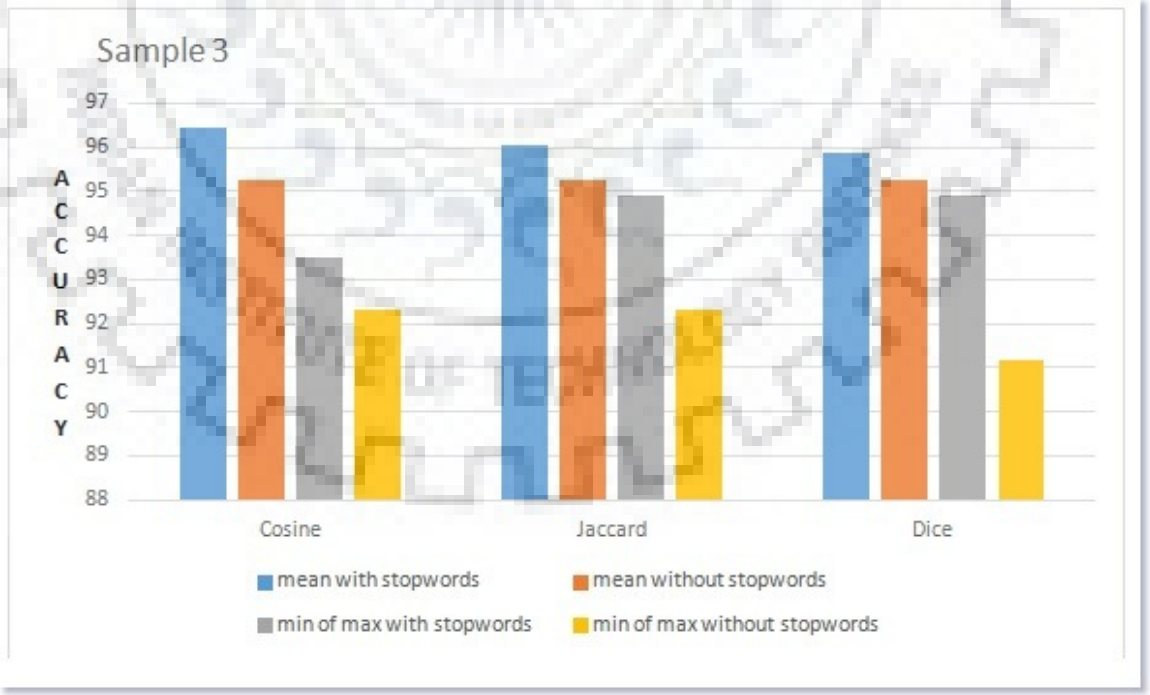


Figure 4.3: *Results for Sample 3 (original captions)*

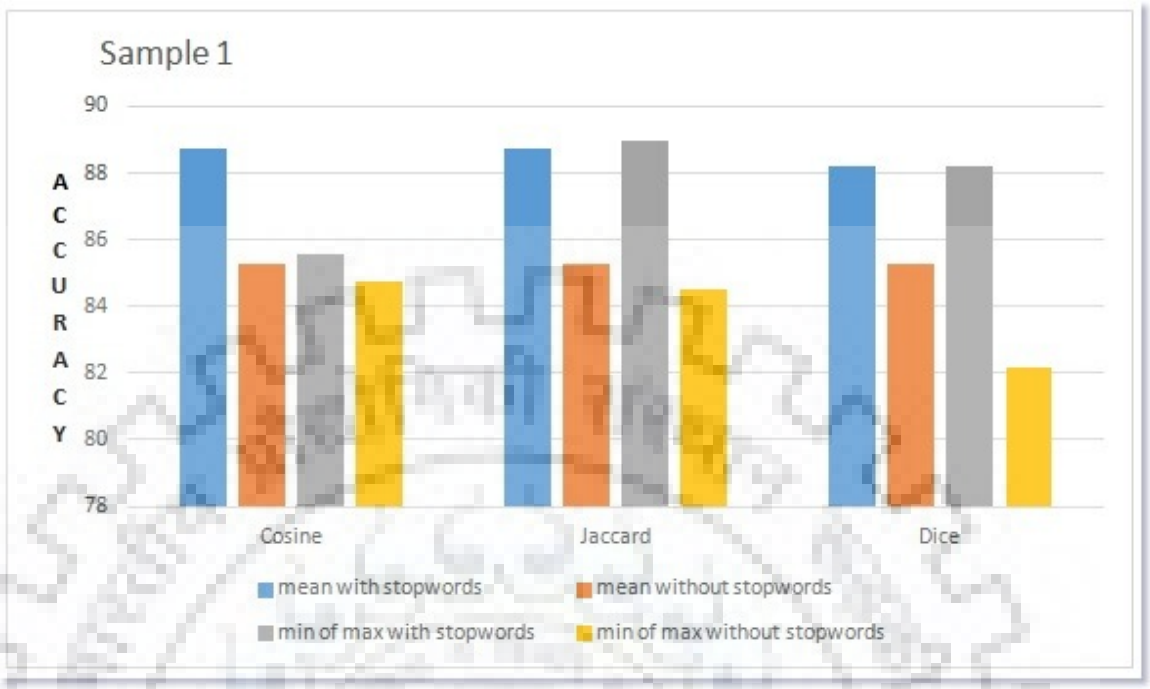


Figure 4.4: *Results for Sample 1 (STT captions)*

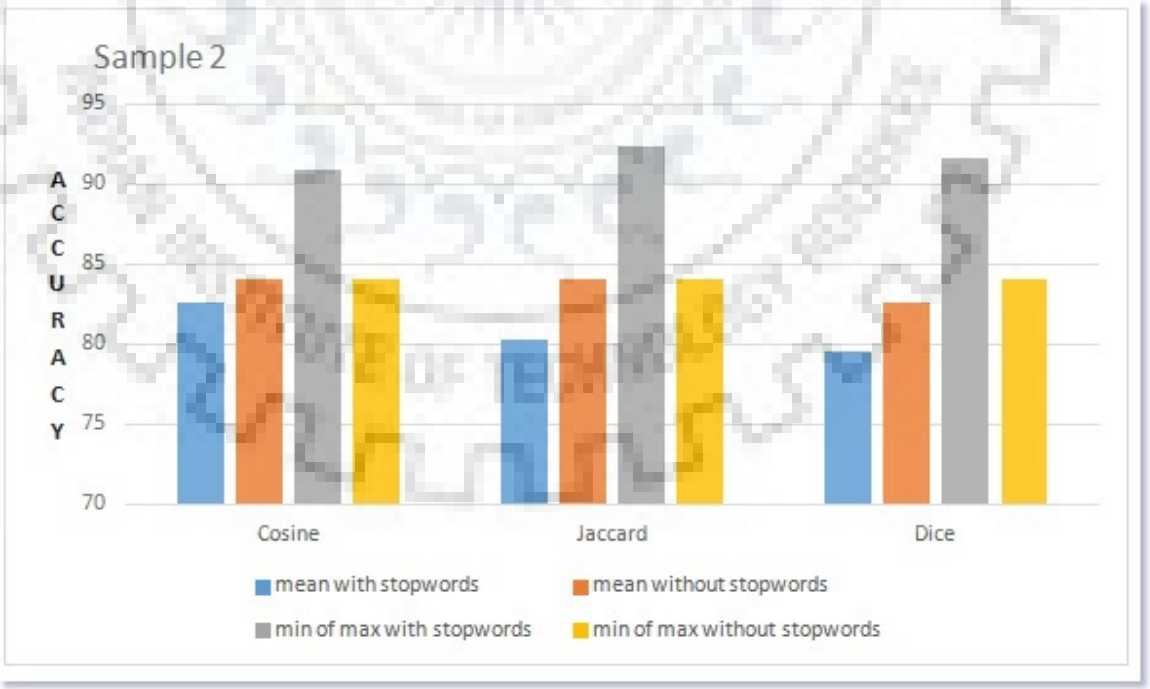


Figure 4.5: *Results for Sample 2 (STT captions)*

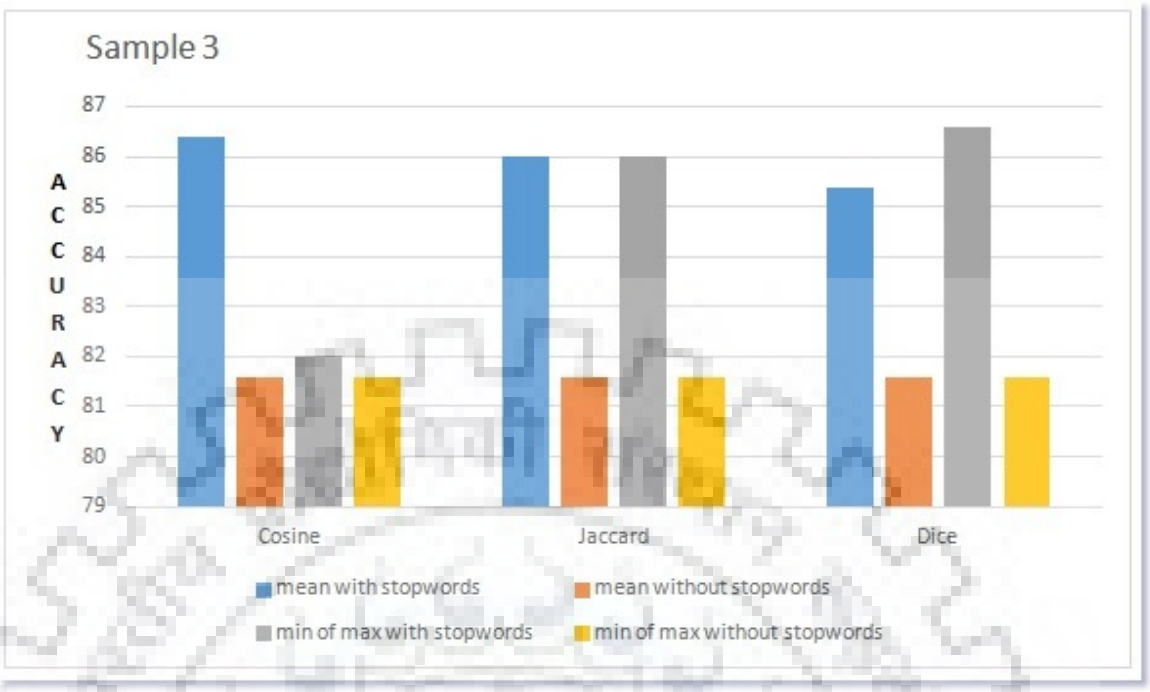


Figure 4.6: *Results for Sample 3 (STT captions)*

4.4 Comparison with Existing Results

In [11] researchers have used word by word comparison where entire document is treated as a single document. The accuracy in such case is generally low. We have also applied word by word technique for finding accuracy. Table 4.10 represents the accuracy in lecture videos and accuracy in movie videos as given in [11]. Instead of relying only on one measure for comparison we have also used different similarity index, In our work we have used cosine, jaccard and dice similarity measure. In [4] authors have used only cosine similarity measure for comparing the accuracy between two sentences. Table 4.11 represents the accuracy when cosine similarity measure is used in [2] and accuracy when cosine similarity measure is used for lecture videos. In Table 4.10 for lecture videos we have given accuracy of captions file that is directly taken from websites of lecture videos. We can clearly see there is an increase of 15 percent in our method.

Table 4.10: *Comparison of word-by-word approach*

Movie Videos(percentage)	**Lecture Videos(percentage)**
54.8	64.3

Table 4.11: *Comparison of cosine similarity approach*

Cooking Videos(percentage)	**Lecture Videos(percentage)**
77.8	96.05

4.5 Discussion

Our system is having little edge over the other systems of caption- script alignment perspective. By our analysis of the results of the experiment done we can get the idea of why the other methods are lagging behind in terms of performance. In case of word by word comparison similarity matrix of words is created, but in our work we have created similarity matrix of sentences. The accuracy in word by word similarity matrix is less as compared to sentence based similarity matrix. Also, the performance reduces because there is no processing of the captions and the scripts file. In our work the processing using NLTK, improves the probability of higher accuracy of alignment. Our proposed caption-script alignment algorithm is achieving higher performance then the existing benchmark algorithms because we are using sentence based approach rather then word based approach, also we have used processed files. The proposed caption-script alignment can be further improved by reducing the information loss during the preprocessing the text, while preprocessing we are trying to remove the not contextual words from our sentences which is sometimes causing a information loss in our sentences. In earlier works authors have used only cosine similarity measure for comparing the similarity between two sentences, but in our work we have also used jaccard and dice similarity measure for finding the similarity. We can see that,in some cases the jaccard and dice similarity measure performs better than cosine

similarity measure. As stated in [3] we can deduce that similarity measure is dependent on the type of content used for measuring the similarity. We can also see from the results that the accuracy in case of YouTube captions is far less compared to the captions taken from the original source of the lecture videos. So in case both the captions are present one should always prefer captions from the source. Also we can deduce that source should always try to provide accurate captions along with their lecture videos, as accurate captions can be useful for further making the videos much more resourceful.

CHAPTER 5

CONCLUSIONS AND FUTURE WORKS

In recent years huge amount of lecture videos are available on-line. The information retrieval field is facing many challenges while dealing with the large amount of video data. People not only watch the videos but they also want some kind of interaction with the videos so that they can interact with the objects appeared in the video, they can find the details of anything with is appeared in the video. For supporting that kind of expectation of the people, we need to transform the videos into interactive videos. So, it has become very important to make these videos resourceful to save users time and energy. Many authors have also applied machine learning techniques for object annotation and index creation of the videos. Here accuracy depends on the technique used for object recognition. Object recognition approach is a challenge in case of lecture videos because there are very few lecture videos with distinct objects. Instead of relying only on machine learning, image processing techniques some researchers use textual meta-data. Another way to annotate videos is using textual data of videos as meta-data. Textual meta-data is important to extract information from the videos. However, only a few videos provide textual meta-data.

In this dissertation we have proposed a system for automatic indexing of lecture videos and analyze resources such as transcripts, lecture videos, and text captions. We have taken scripts and captions from lecture videos, and also captions generated from YouTube's auto-caption

feature. Based on the analysis, a similarity matrix is constructed using sentences in captions and script. The DP algorithm is applied to find optimal paths in the matrix. To evaluate the proposed method, alignment accuracy is measured. According to the experimental results, the proposed method provides an average increase of 15percent in alignment accuracy in comparison to the existing methods. For comparing the similarity between two sentences we have used different similarity measure these are cosine similarity, jaccard similarity and dice similarity.

This work proposes a method for automatically indexing lecture videos. We have also proposed a new equation (3.5) to calculate the threshold value for the similarity measures. Also from our experiments we can state that, the value of alpha parameter used in [2] is dependent upon the equation used for calculation of threshold value. Based on our experiments and as mentioned by authors in [3] we have concluded that, jaccard and dice similarity measure performs better than cosine when the sentences have similar structure but different semantics. This is contrary to the approach followed in [2] where only cosine similarity index is used for measuring similarity of the sentences. As stated, the proposed method shows higher alignment performance as compared to existing approaches. As, per our knowledge this is the first proposal for lecture videos in the textual domain where alignment can be used for indexing. However, alignment accuracy can be improved by applying natural language processing (NLP) techniques to the problem of synonyms [7] and enhancing the accuracy of terms in generated text captions. In addition, aligned information can be used to annotate the scripts and its related information in lecture videos. We can also design a user interface by which users can go to the desired part of the lecture videos by using functions on the interface. With these features, the proposed method can facilitate the interactivity between users and lecture videos. In addition, the interactive lecture video services can also be provided to the tutor web site. The works can be extended to other domain of the videos such as news videos or entertainment videos. News videos and entertainment videos can also be made interactive. In addition the caption based approach can also play very useful

role in searching through the videos.

REFERENCES

[1] Z. Xiong, X. S. Zhou, Q. Tian, Y. Rui, and H. TS, "Semantic retrieval of video - review of research on video retrieval in meetings, movies and broadcast news, and sports," *IEEE Signal Processing Magazine*, vol. 23, pp. 18–27, March 2006.

[2] K. J. Oh, M. D. Hong, S. Y. Sim, and G. S. Jo, "Automatic indexing of cooking video by using caption-recipe alignment," in *2014 International Conference on Behavioral, Economic, and Socio-Cultural Computing (BESC2014)*, pp. 1–6, Oct 2014.

[3] S. M. Saad and S. S. Kamarudin, "Comparative analysis of similarity measures for sentence level semantic measurement of text," in *2013 IEEE International Conference on Control System, Computing and Engineering*, pp. 90–94, Nov 2013.

[4] L. Ballan, M. Bertini, A. Del Bimbo, L. Seidenari, and G. Serra, "Event detection and recognition for semantic annotation of video," *Multimedia Tools and Applications*, vol. 51, no. 1, pp. 279–302, 2011.

[5] J. Wu and M. Worring, "Efficient genre-specific semantic video indexing," *IEEE Transactions on Multimedia*, vol. 14, pp. 291–302, April 2012.

[6] T. Cour, C. Jordan, E. Miltsakaki, and B. Taskar, *Movie/Script: Alignment and Parsing of Video and Text Transcription*, pp. 158–171. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008.

[7] R. Ronfard and T. T. Thuong, "A framework for aligning and indexing movies with their

script," in *Multimedia and Expo, 2003. ICME '03. Proceedings. 2003 International Conference on*, vol. 1, pp. I–21–4 vol.1, July 2003.

[8] R. Turetsky and N. Dimitrova, "Screenplay alignment for closed-system speaker identification and analysis of feature films," in *2004 IEEE International Conference on Multimedia and Expo (ICME) (IEEE Cat. No.04TH8763)*, vol. 3, pp. 1659–1662 Vol.3, June 2004.

[9] Pooja and R. Dhir, "Video text extraction and recognition: A survey," in *2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, pp. 1366–1373, March 2016.

[10] S. B. Park, H. N. Kim, H. Kim, and G. S. Jo, "Exploiting script-subtitles alignment to scene boundary dectection in movie," in *2010 IEEE International Symposium on Multimedia*, pp. 49–56, Dec 2010.

[11] A. Lambert, M. Guegan, and K. Zhou, "Scene reordering in movie script alignment," in *2013 11th International Workshop on Content-Based Multimedia Indexing (CBMI)*, pp. 213–218, June 2013.

[12] K.-J. Oh, M. Hong, U.-N. Yoon, and G. S. Jo, "Automatic generation of interactive cooking video with semantic annotation," vol. 22, pp. 742–759, 01 2016.

[13] X. C. Yin, Z. Y. Zuo, S. Tian, and C. L. Liu, "Text detection, tracking and recognition in video: A comprehensive survey," *IEEE Transactions on Image Processing*, vol. 25, pp. 2752–2773, June 2016.

[14] P. Achananuparp, X. Hu, and X. Shen, *The Evaluation of Sentence Similarity Measures*, pp. 305–316. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008.

[15] A. Huang, "Similarity measures for text document clustering," pp. 49–56, 2008.

[16] X. Liu, Y. Zhou, and R. Zheng, “Sentence similarity based on dynamic time warping,” in *International Conference on Semantic Computing (ICSC 2007)*, pp. 250–256, Sept 2007.

[17] B. D. Homer and J. L. Plass, “Level of interactivity and executive functions as predictors of learning in computer-based chemistry simulations,” *Computers in Human Behavior*, vol. 36, pp. 365 – 375, 2014.

[18] J. N. O. Dag, B. Regnell, P. Carlshamre, M. Andersson, and J. Karlsson, “Evaluating automated support for requirements similarity analysis in market-driven development,” in *In Seventh International Workshop on Requirements Engineering: Foundation for Software Quality (REFSQâĂŹ01*, 2001.

Shubhangi Report

ORIGINALITY REPORT

6%	1%	6%	1%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

Oh, Kyeong-Jin, Myung-Duk Hong, Sang-Yong Sim, and Geun-Sik Jo. "Automatic indexing of cooking video by using caption-recipe alignment", 2014 International Conference on Behavioral Economic and Socio-Cultural Computing (BESC2014), 2014. Publication	3%
Saad, Sazianti Mohd, and Siti Sakira Kamarudin. "Comparative analysis of similarity measures for sentence level semantic measurement of text", 2013 IEEE International Conference on Control System Computing and Engineering, 2013. Publication	1%
Sudhakaran, Periakaruppan, Shanmugasundaram Hariharan, and Joan Lu. "Classifying Product Reviews from Balanced Datasets for Sentiment Analysis and Opinion Mining", 2014 6th International Conference on Multimedia Computer Graphics and Broadcasting, 2014.	<1%

Publication

Submitted to University of Warwick <1%
Student Paper

T.T. Thuong. "A framework for aligning and indexing movies with their script", 2003 International Conference on Multimedia and Expo ICME 03 Proceedings (Cat No 03TH8698) ICME-03, 2003 <1%
Publication

Submitted to Vrije Universiteit Amsterdam <1%
Student Paper

www.rroij.com <1%
Internet Source

www.fdot.gov <1%
Internet Source

Exclude quotes	On	Exclude matches	< 10 words
Exclude bibliography	On		