Text Preprocessing on Extracted Text from Audio/Video using R

Text Preprocessing on Extracted Text from Audio/Video using R

B. K. Poornima
Department of Computer Application
Government Arts College (A)
Salem -7
bkpoornimoorthy23@gmail.com

D. Deenadayalan
Department of Computer Application
Government Arts College (A)
Salem -7
deena.duraisamy@gmail.com

A. Kangaiammal
Department of Computer Application
Government Arts College (A)
Salem-7
indurath2002@yahoo.com

***Abstract-* E-learning is an evolving new education approach that increases learning experiences by integrating multimedia and network technologies. As an integrated part of e-learning, the sources of e-learning content are text, audio, video, images and animations. Most of the instructional content is in the form of lecture videos. Effective use of these videos, however, remains a challenging task. To overcome this issue, the video content has to be analyzed effectively. For analyzing the video content, the audio part of the video can be converted to text. The textual form of content enables the content developers in analyzing the content in a manner with reduced time and space. The accuracy of content analysis may be improved in the textual form of the content than other forms. Therefore first the text has to be converted from audio and video using the existing tools and techniques such as Google Docs, Braina, Apple Dictation, Dragon Naturally Speaking, etc. As the converted text is with errors they may be preprocessed before it is used. Hence this paper uses the tool R to preprocess the text obtained from audio and video through Google Docs open source. This preprocessing paves a way to reduce the error rate. It is observed that the improvement of 0.94% precision accuracy of converted text; while the improvement of recall accuracy is 0.91% after preprocessing. Therefore, the false rate has reduced after preprocessing. Hence preprocessing reduces the error rate around 1% on the converted text from audio and video.**

Keywords: E-learning, Multimedia content, Google Docs, Preprocessing

1. INTRODUCTION

E-learning is the use of internet technologies to enhance knowledge and performance. The rapid growth of computer and Internet technologies has made E-Learning become an important learning method. The learner's needs for multimedia instructional material in e-learning has increased recently and such content has been included to attract a learner's attention and interests (Parlakkilic& Karslioglu, 2013).

Recent advances in multimedia technology have led to tremendous increases in the available volume of video data, thereby creating a major requirement for efficient systems to manage such huge data volumes (Irfan Mehmood et al, 2016). Multimedia content mainly contains video, audio, movies, animation, images and graph. The development of multimedia e-learning content has been employment intensive process, requiring a team of web designers and developers responsible for the technical development of these resources, thereby limiting its widespread implementation. The powerful and expressive multimedia that can capture and present information, instructional videos are extensively used in e-learning (Dongsong Zhang, Jay F., & Nunamaker, 2004).Therefore, the learners can easily learn the video contents. Learners aim to retrieve videos of interest having content which is relevant to their need(H. Bhaumik, 2016).Video content is rich and expressive. Analyzing the video content is technically difficult to produce results and also take more time for access. Text content can be easily analyzed and results may be produced in less time.

Google Apps is a collection of web-based programs and data storage that run in a web browser, without requiring users to buy or install software. Users can simply log in to the service to access their files and the tools to manipulate them. The offerings include communication tools (Gmail, Google Talk, and Google Calendar), productivity tools (Google Docs: text files, spreadsheets, and presentations), a customizable start page (iGoogle), and Google Sites (to develop web pages). Google Apps allows institutions to use their own domain name with the service and to customize the interface to reflect the branding of that institution. In this way, an educational institution can offer the functionality of Google Apps in a package (and with a URL) that is familiar and comfortable to constituents. Google Docs is a web-based word processing application that allows users to create, view and share documents, presentations, and spreadsheets over the Internet (Mohammed Al-Zoube & Baha Khasawneh , 2010). Google Docs App is free software and it is only used by online that is a main advantage for converting a video content to text content. Using Google Docs application is effective to convert the audio text from audio and video content to text form.

Such conversion cannot be accurate and error free; hence it needs to be preprocessed to identify the presence of noise, locate errors and measure the error rate and thereby correct the errors and noise. Appropriate layers of tools and techniques may be applied to carry out such preprocessing.

Preprocessing is an important task and critical step in Text mining, Natural Language Processing (NLP) and Information Retrieval (IR). Data preprocessing is used for extracting interesting and non-trivial and knowledge from unstructured text data. Four common preprocessing steps including tokenization, stop-word removal, lowercase conversion, and stemming are considered within the scope of this paper (X.S. Hua, L. Wenyin, & H.J. Zhang, 2004).

In Preprocessing, i) tokenization is the procedure of splitting a text into words, phrases, or other meaningful parts, namely tokens. In other words, tokenization is a form of text segmentation. Typically, the segmentation is carried out considering only alphabetic or alphanumeric characters that are delimited by non-alphanumeric characters (e.g., punctuations, whitespace). ii) Stop-words are the words that are commonly encountered in texts without dependency to a particular topic (e.g., conjunctions, prepositions, articles, etc.). iii) Another widely used preprocessing step is lowercase conversion. Since uppercase or lowercase forms of words are assumed to have no difference, all uppercase characters are usually converted to their lowercase forms. iv) The aim of stemming is to obtain stem, or root, forms of derived words. Since derived words are semantically similar to their root forms, word occurrences are usually computed after applying the stemming on a given text (Alper Kursat Uysal & Serkan Gunal, 2014). Using these preprocessing methods to identify the presence of noise, locate errors and measure the error rate and thereby correct the errors and noise to make the content suitable for content analysis.

R is a widely used analysis environment for scientific computing and visualization, such as statistics, data mining, bioinformatics, and machine learning. Currently, over 5000 packages are included in the repositories of the Comprehensive R Archive Network. Every package submitted into the repositories is checked to meet some quality standards, such as representative documentation and running on any operating systems (e.g., MS Windows, Mac OS X, Linux) (Muenchen, 2013).

R is one of the tools to preprocess the text data in an efficient manner. For preprocessing the converted text from audio/video, text mining tasks need to be performed. The primary package for Text Mining is tm (Graham Williams, 10th January 2016). The main structure for managing text documents in tm is a so-called Corpus, representing a collection of text documents. A corpus is an abstract concept, and there exist several implementations in parallel. Once the corpus is formed, the documents in the corpus are to be typically modified by the preprocessing techniques such as stemming, stopword removal, convert text to lowercase and eliminating whitespace. In tm, all these functionalities are subsumed into the concept of a transformation. Transformations are done via the tm_map() function which applies (maps) a function to all elements of the corpus. Basically, all transformations work on single text documents and tm_map() just applies them to all documents in a corpus (Ingo Feinerer, 2015).

2. REVIEW OF LITERATURE

E-Learning is an application which revolutionizes traditional education, as it could provide faster access to materials at minimal cost. E-learning is an application where learner can gain knowledge by accessing materials. It is one of the way in which a learner can have access to a topic in different forms of learning materials like text, pdf, PowerPoint presentations, audio, video, etc. (Kaladevi A, Padmavathy S, Kangaiammal A., & S.Theetchenya, 2013)

2.1. Multimedia E-learning

The use of multimedia based teaching material enhances students learning and increase productivity. It is therefore important for faculty, not only to understand the concepts behind the development of multimedia content but to also have a good grasp of how to implement some of the processes involved with courseware production (Dongsong Zhang , Jay F., & Nunamaker, 2004). The digital video archives are challenging problem and rapidly gaining widespread research and commercial interest. Learners can quickly locate their interested content in a colossal volume of video data; the extraction of content descriptive features is required. However, it is well recognized that low-level features as measures of color, texture and shape (S. Antani, R. Kasturi, & R. Jain, 2002) are not enough for uniquely discriminating across different video content. Extracting more descriptive features and higher level entities as text is more important(M. Lyu, J. Song, & M. Cai, 2005). Among all these features, text is most reliable for this purpose especially the audio will be converted to textual form, if this audio content converted into textual form automatically, they would be a valuable source of high-level semantics for analyzing the video content.

Ankur et al attempt to classify the videos present on YouTube on the basis of Tags and Focal Points, but they did not consider the texts which are presented in the video (Ankur, 2016). (Bassem 2013) mentions that Video-text localization and identification approach which proceeds in two main types as text region localization and text region classification. Here they focus only on the superimposed text presented in the video (Bassem , 2013).

Ying Li et al (2006) proposed Video abstraction is a technique that abstracts video content and represents it in a compact manner. There are basically two types of video abstraction: video summarization and video skimming (Ying Li, 2006). Video summarization is a process that selects a set of salient images called key frames to represent the video content. Video skimming represents the original video in the form of a short video clip. But the entire text was not considered for extracting the information presented in the video. The voice part of video has to be converted into textual form thereby to get the entire information about the video content.

2.2. Preprocessing of Text Document

Preprocessing is an important task and critical step in Text mining, Natural Language Processing (NLP) and Information Retrieval(IR). In the area of Text Mining, data preprocessing used for extracting interesting and non-trivial and knowledge from unstructured text data. The preprocessing techniques such as Tokenization, Stop word removal, lowercase conversion and Stemming are used for the text documents.

Stop words removal technique is to remove the stopwords such as prepositions, articles, pronouns, etc. (that does not give the meaning of the documents i.e the, in, a, an, with). Removing stopwords reduces the dimensionality of term space.

Tokenization is to identify the meaningful keywords. The inconsistency can be different number and time formats. Challenges in tokenization depend on the type of language. Languages such as English and French are referred to as space delimited as most of the words are separated from each other by white spaces. Tokenization is also affected by writing system and the typographical structure of the words. Another problem is abbreviations and acronyms which have to be transformed into a standard form.

Lowercase conversion is for a word that appears exactly the same every time it appears. The purpose of this method used for all text content is that all text is in the form of lowercase and so easy to analyze the content.

Stemming is used to identify the root/stem of a word. The purpose of this method is to remove various suffixes, to reduce the number of words, to have accurately matching stems, to save time and memory space.

There are mainly two errors in stemming that are over stemming and under stemming. Over-stemming is when two words with different stems are stemmed to the same root. This is also known as a false positive. Under-stemming is when two words that should be stemmed to the same root are not. This is also known as a false negative. There are three types of stemming algorithms used to reduce the size of the text document. They are Successor Variety, N-gram stemmers and Affix Removal Stemmers (Vairaprakash Gurusamy & Subbu Kannan, 2014).

2.2.1 Successor Variety

Successor variety stemmers are based on the structural linguistics which determines the word and morpheme boundaries based on distribution of phonemes. Successor variety of a string is the number of characters that follow it in words in some body of text. For example consider a body of text consisting of following words Stemming and Stemmer (Vairaprakash Gurusamy & Subbu Kannan, October 2014).

2.2.2 N-Gram stemmers

It is called as shared digram method. Digram is a pair of consecutive letters. This method is called n-gram method since trigram or n-grams could be used. In this method association measures are calculated between the pairs of terms based on shared unique digram.
For example: consider two words Stemming and Stemmer

Stemming → stteem mm mi in ng
Stemmer → stteem mm me er

In this example the word stemming has 7 unique digrams, stemmer has 6 unique digrams, these two words share 5 unique digramsst, te, em, mm, me. Once the number of unique digrams is found then a similarity measure based on the unique digrams is calculated using dice coefficient. Dice coefficient is defined as in the equation (1).

$$S=2C/ (A+B) \quad (1)$$

Where C is the common unique digrams, A is the number of unique digrams in first word; B is the number of unique digrams in second word. Similarity measures are determined for all pairs of terms in the database, forming a similarity matrix. Once such a similarity matrix is available, the terms are clustered using a single link clustering method (S. Vijayarani, Ms. J. Illamathi, & Ms. Nithya, 2016).

2.2.3 Affix Removal Stemmers

Affix removal stemmers removes the suffixes or prefixes from the terms leaving the stem. One of the examples of the affix removal stemmer is one which removes the plurals form of the terms. Some set of rules for such a stemmer are as follows

a) If a word ends in “ies” but not “eies” or “aies ” Then “ies” -> “y”
b) If a word ends in “es” but not “aes”, or “ees” or “oes” Then “es” -> “e”
c) If a word ends in “s” but not “us” or “ss ” Then “s” -> “NULL”

Using these efficient preprocessing techniques eliminates noisy from text data, later identifies the root word for actual words and reduces the size of the text data. This improves performance of the IR system (Vairaprakash Gurusamy & Subbu Kannan, 2014).

From the above literature, tags and focal points are used for text conversion. However, the entire text cannot be extracted from the video. Only the video author, video link, video category and video tags are not enough for the text classification. Another method of video extraction is video-text localization and identification approach that focuses only the superimposed text which is present in the video which is not enough for the text classification. The video abstraction is a technique that abstracts video content and represents it in a compact manner. Here video summarization and video skimming are the two methods of video abstraction that are only represented as key frames and form of short video clip. Here the entire text is not extracted from the videos. Moreover, video includes actions, postures and gestures to cover more text content which cannot be considered this way.

It is understood that from the above literature, the text form of video is a partial content only. However, the analysis is relatively possible with text than video. Hence, this paper aims to bring out an approach to extract the superimposed audio from video content into text for content analysis purpose.

3. PROPOSED WORK

Methods to overcome the issues indicated from the literature are the need of the hour. However, the proposed method is mainly focusing to extract the superimposed text in the form of audio in a video and apply some preprocessing to reduce the error rate and to make it useful for content analysis. Though the entire text should be extracted from the video for the content analysis, the superimposed audio from video alone is converted to text using any one of the many existing tools such as Google Docs, Braina, Apple Dictation, Dragon Naturally Speaking, etc. This work uses Google Docs app. From Google Docs app, Google Voice Search, a Google product that allows users to use Google Search by Speaking on a mobile phone or computer is used. In Google Docs, a popular service for managing documents online. Google Docs supports typing, editing and formatting via voice commands. It also helps to capture ideas, compose a letter or even write the novel without touching the keyboard.

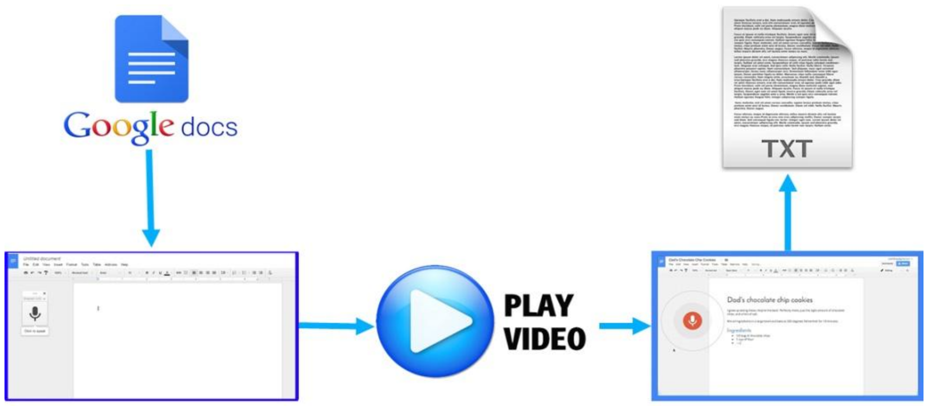


Figure 1. Video to Text Conversion

Figure 1 presents that the workflow of the conversion of video to text. Figure-1 portrays that the superimposed audio from the video must be extracted as text using voice typing contained in the Google Docs App. The output of conversion is multiple text documents. To open up a Google Docs document, click on Tool->Voice Typing. Click on it to activate the voice typing mode. Then start playing the video which want to extract the text from the video. Format the document's text with any of the video. Finally save the text document which is extracted from video as text file.

This text document is not ready for content analysis because it cannot be accurate and error free. Preprocessing techniques are used to identify the presence of noise, locate errors and measure the error rate and thereby correct the errors and noise. Text preprocessing is an essential part of any NLP (Natural Language Processing) system, since the characters, words, and sentences identified at this stage are the fundamental units passed to all further processing stages, from analysis.

R is an important tool for text mining, statistical tests, content analysis, etc. R is mainly used for large amount of documents. First load the R package for text mining and then load texts into R. Then all documents loaded properly, go on to preprocessing texts. Using R tool here multiple documents are preprocessed using several preprocessing techniques as mentioned.

Figure 2 shows that preprocessing of multiple number of converted text documents are fed as input. The first preprocessing technique Stopwords Removal is to remove the stopwords. The second preprocessing technique Tokenization is the procedure of splitting a text into words, phrases, or other meaningful parts, namely tokens. Typically, segmentation is carried out considering only alphabetic or alphanumeric characters that are delimited by non-alphanumeric characters (e.g., punctuations, whitespace). The third preprocessing technique is Lowercase Conversion. Uppercase or lowercase forms of words are assumed to have no difference; all uppercase characters are usually converted to their lowercase forms. The fourth preprocessing technique Stemming is used to identify the root/stem of a word. For example, the words connect, connected, connecting, connections all can be stemmed to the word "connect".

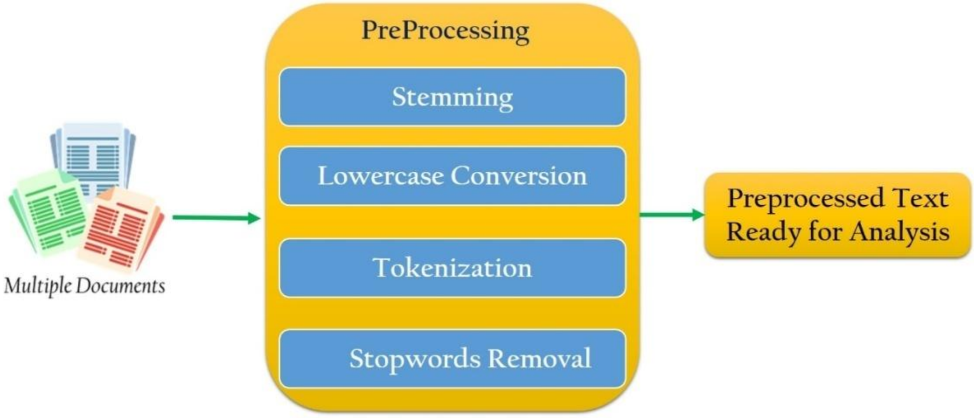


Figure 2. Text Preprocessing for Content Analysis

Once the preprocessing as per the above four techniques have been carried out, the final text outcome is to be used for content analysis. Before content analysis Validate the text obtained for error rate and correctness otherwise apply yet another preprocessing for further noise removal. That is left as the further extension of this work.

4. EXPERIMENTAL WORK

It is obvious that the text extracted from audio/video using Google Docs Application is prone for errors and hence the need for preprocessing. The preprocessing is carried out using the tool R that includes advanced statistical analyses and supports highly unstructured data.

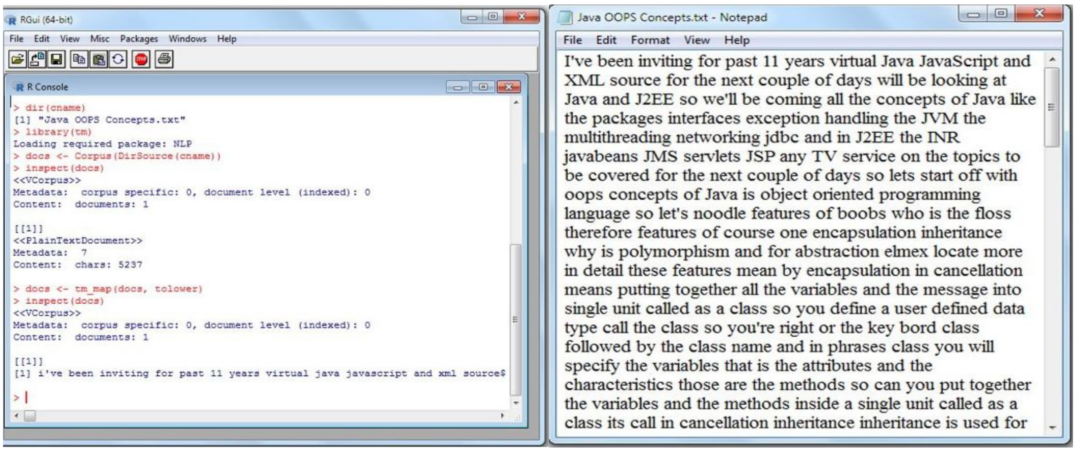
The experimentation process deals with five different video for different playtime ranging from 3.47 to 11.23 minutes (3.47, 5.36, 7.56, 10, 11.23). The test video containing929 word occurrence that plays for a maximum time of 11.23 minutes is the largest sample chosen. Similarly, 3.47 minutes length video includes 328 words; 5.36 playtime includes 452 words; 7.56 minutes video with 657 words; and 10 minutes video with 822 words are part of the experiment. For a quantitative evaluation, the correct text, the detected text and the ground-truth text are defined. Thus, a set of quantitative evaluation measurements Recall, Precision and False rate are computed for each of the experiment. Recall rate evaluates to how much percentage of all ground-truth video text are correctly detected. The Precision rate evaluates to how much percentage of the detected video text are correct. False rate evaluates to how much percentage of the detected videotext are wrong. The formula for these measurements is presented in Figure 3 (X. Hua, P. Yin, & H.J. Zhang, 2002) (X.S. Hua, L.Wenyin, & H.J. Zhang, 2004).

As a sample, the original video text and the preprocessed text for 3.47 minutes playtime is presented in Figure 4.

As shown in Figure 4, the original text is taken as input to R tool and preprocessed. This process undergoes the above mentioned all the four techniques. The outcome statistics of the processing is in ‘Preprocessed Text Statistics in R’ part of Figure 4. It is observed that the total number of characters is reduced from the original text. In this outcome, the results of precision and recall value are increased and the false rate value is reduced. As the total number of characters is reduced, the time taken and the storage space required also has reduced.

Measurements	Values
Recall	Number of Correctly Detected Text Divided by Number of All Ground-Truth Text
Precision	Number of Correctly Detected Text Divided by Number of Detected Text
False Rate	Number of Wrongly Detected Text Divided by Number of Detected Text

Figure 3. Formulas for Evaluation Measurements



Original Text Preprocessed Statistics in R

Figure 4. Text Preprocessing Statistics in R Tool

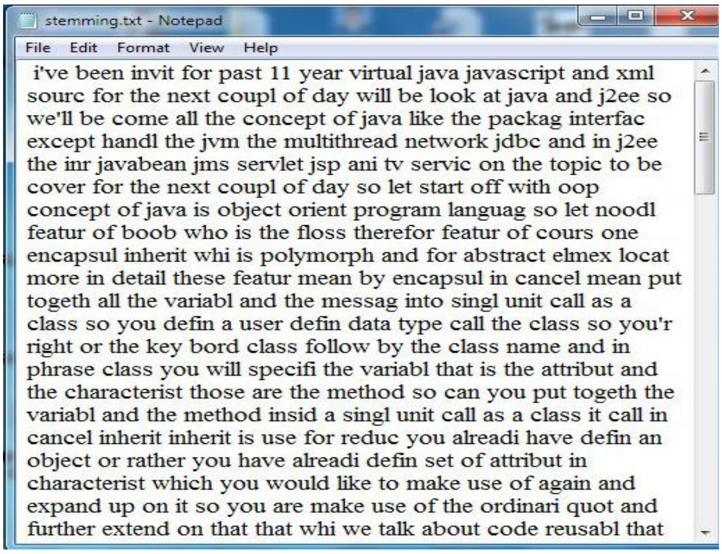


Figure 5. Preprocessed Text after Stemming

Figure 5 presents the original text is taken as input in R tool and preprocessed using Stemming technique. After removal of the stems, the total number of original text is reduced as clearly shown in 'After Stemming' part of Figure 5. This method is mainly used to save time and memory space.

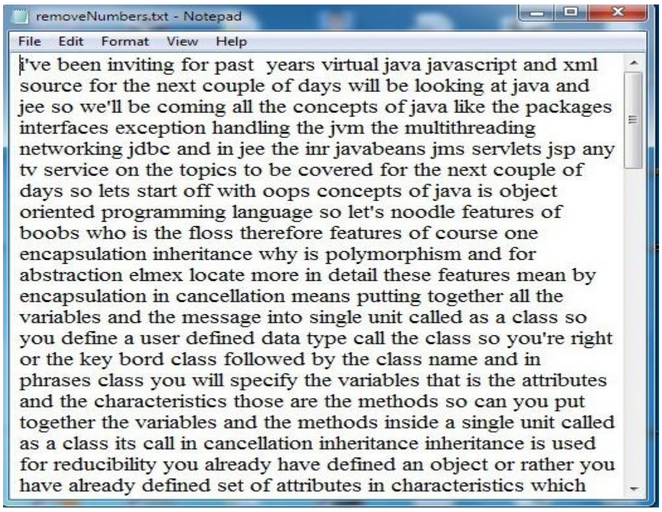


Figure 6. Preprocessed Text after Removal of Numbers

Figure 6 indicates the original text as input and preprocessed using Removal of Numbers. After removal of the numbers, the total number of original text is reduced to little extent atleast as it depends on the nature of content and the presence of numbers. It is seen from the ‘Removal of Numbers’ part of Figure 6. This method of preprocessing is mainly to save time and memory space.

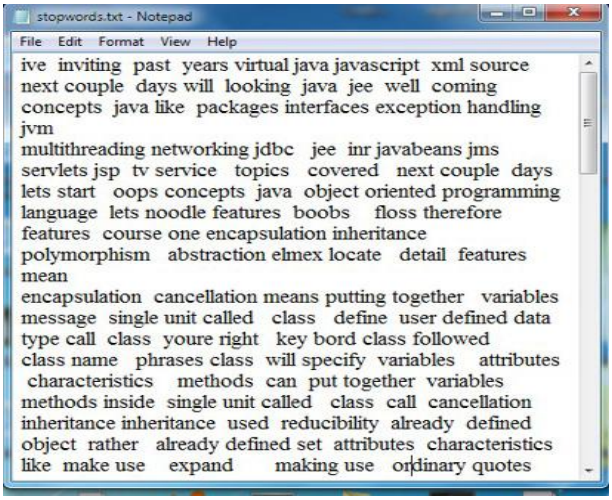


Figure 7. Preprocessed Text after Stopwords Removal

After the removal of the stopwords, the total number of characters is reduced as shown in ‘Stopwords Removal’ part of Figure 7. This method is mainly used to save time and memory space.

5. RESULTS AND DISCUSSION

5.1. Manually Converted Text before Preprocessing

The superimposed audio of the video is manually listened and converted to text. Table 1 shows that the results for the manually converted text is with Precision and Recall value is 100% for all video files while the measurement of false rate is 0.

Table : 1 Quantitative Evaluation Measurements for Manually Converted Correct Text

Topic / Measures	JAVA Introduction	JAVA OOPS Concept	JAVA Control Structure	JAVA Inheritance	JAVA Beans
	Play Time 11.23 min	Play Time 7.56 min	Play Time 10 min	Play Time 5.36 min	Play Time 3.47 min
Recall	100%				
Precision	100%				
False Rate	0				

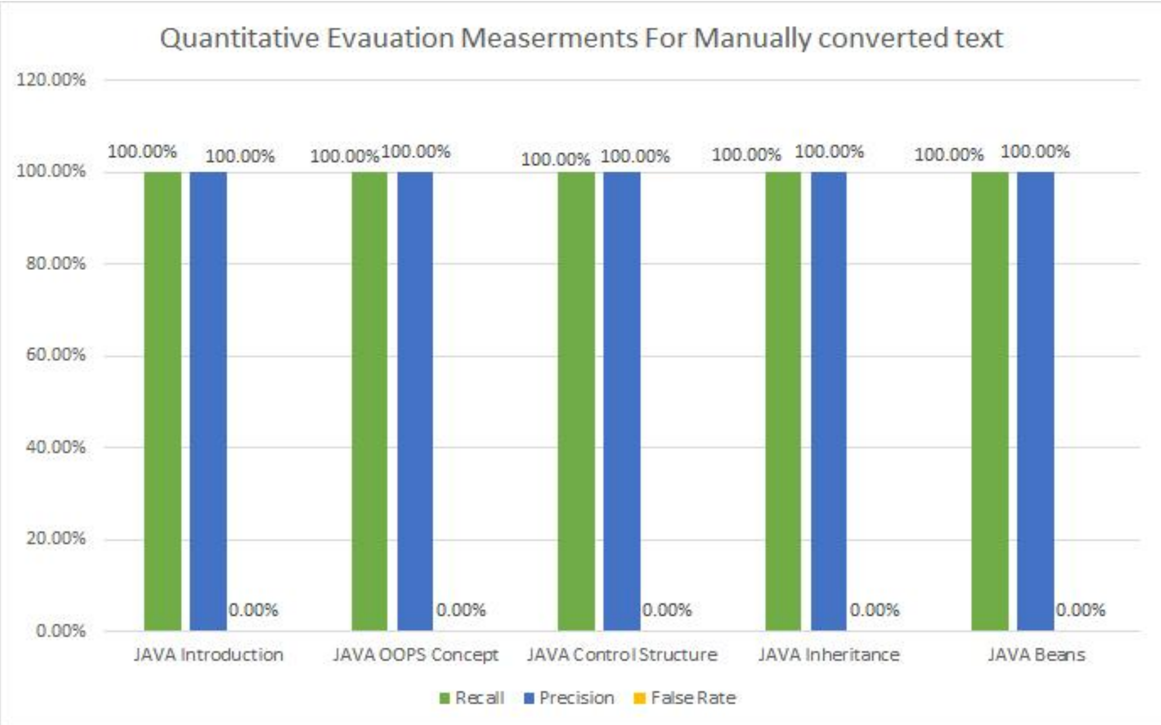


Figure 8. Quantitative Evaluation Measurements of Manually Converted Text

5.2. Comparison of Manually Converted Text before and after Preprocessing

In Preprocessing the manually converted text, the total number of word count only be reduced. The error values do not occur during the preprocessing methods.

Table : 2 Comparison of Manually Converted Text before and after Preprocessing.

Topic / Word Count	Java Introduction		Java OOPS Concept		Java Control Structures		Java Inheritance		Java Beans	
	Play Time 11.23 min	Play Time 11.23 min	Play Time 7.56 min	Play Time 7.56 min	Play Time 10 min	Play Time 10 min	Play Time 5.36 min	Play Time 5.36 min	Play Time 3.47 min	Play Time 3.47 min
Detected Words	902	894	611	604	758	752	428	414	312	305
Correctly detected words	902	894	611	604	758	752	428	414	312	305
Wrongly detected Words	0	0	0	0	0	0	0	0	0	0

Table 2 shows that the five different topics for the five different play time video is manually converted to text. The manually converted text before and after preprocessing in terms of word counts are shown as the total number of detected words, the number of correctly detected words and the number of wrongly detected words. The number of wrongly detected words is zero. The total number of detected words and the number of correctly detected words are same for the manually converted text. The word count has significantly reduced after preprocessing the manually converted text. As the word count is reduced then the time taken and storage space required is also reduced. Therefore the time, space and processing requirement for further processing will be eased.

5.3. Quantitative Evaluation Measurements of Automatic Conversion before and after Preprocessing

Similar processing is carried out for the text converted from Google Docs application is considered.

Table : 3Quantitative evaluation measurements of automatic conversion before and after preprocessing

Topic	Java Introduction		Java OOPS Concept		Java Control Structures		Java Inheritance		Java Beans	
	Before%	After%	Before%	After%	Before%	After%	Before%	After%	Before%	After%
Measures	Play Time 11.23 min	Play Time 11.23 min	Play Time 7.56 min	Play Time 7.56 min	Play Time 10 min	Play Time 10 min	Play Time 5.36 min	Play Time 5.36 min	Play Time 3.47 min	Play Time 3.47 min
Recall	88.80	89.73	92.39	93.46	90.27	91.36	93.58	94.69	94.21	95.12
Precision	92.14	93.11	94.99	96.09	93.22	94.35	95.49	96.61	96.26	97.20
False Rate	7.86	6.89	5.01	3.91	6.78	5.65	4.51	3.39	3.74	2.80

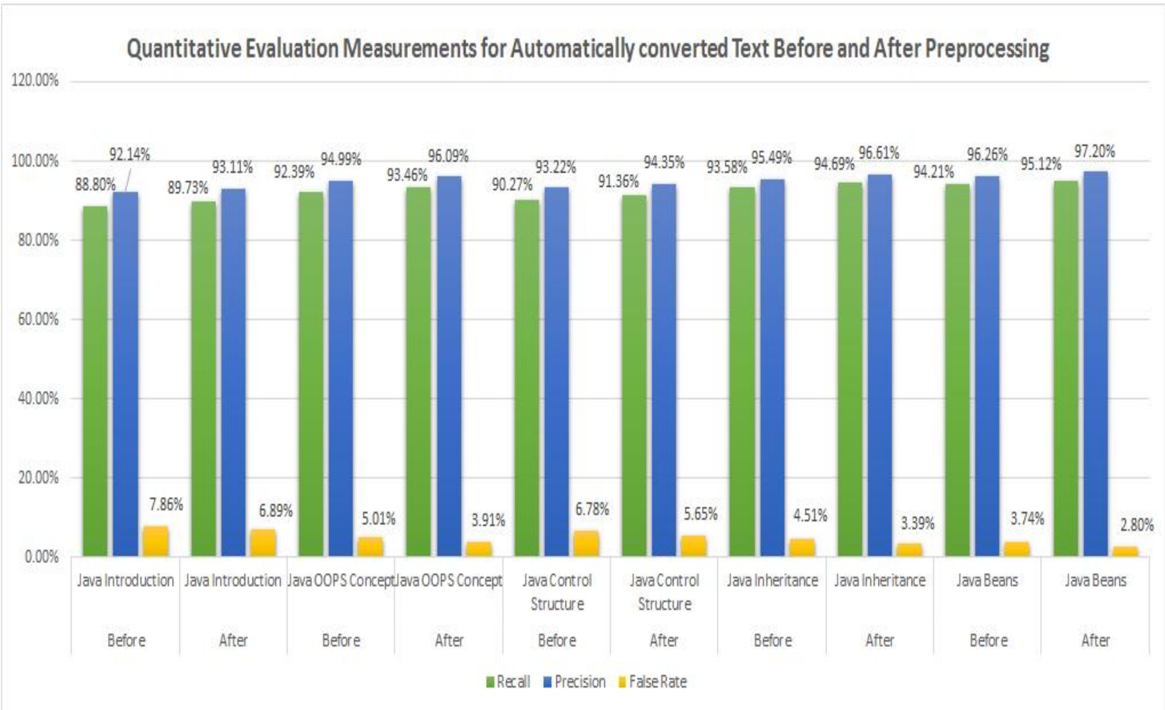


Figure 9. Quantitative Evaluation Measurements of Automatic Conversion before and after Preprocessing

Table 3 and Figure 9 shows that the results for the automatically converted text from Google Docs before and after preprocessing. The highest precision value is 96.26% and the highest recall value is 94.21% for the lowest play time video for before preprocessing. The lowest play time video contains less error, so that the accuracy of the precision and recall value is increased for the lowest play time video before preprocessing. The measurement of shortest false rate value is 3.74% for the shortest play time video for before preprocessing. The measurement of highest precision value is 97.20% and the highest recall value is 95.12% for the lowest play time video after preprocessing. Lowest play time video with less errors, the accuracy of the precision and recall value are increased for the lowest play time video for the text after preprocessing. The measurement of shortest false rate value is 2.80% for the shortest play time video for the text after preprocessing. It is understood from the experimental results that when there is an increase in playtime, there is an increase in false rate is also. The vice-versa is also true.

The results of the automatic converted text before preprocessing and the automatic converted text after preprocessing are compared using the measurements of Precision, Recall and False Rate value. The precision value is 96.26% for before preprocessing and the precision value is 97.20% for after preprocessing. The difference between the precision value before and after preprocessing is 0.94%. The results show that the accuracy of the automatic converted text after preprocessing is 0.94% increased than the automatic converted text before preprocessing. The recall value is 94.21% for before preprocessing and the recall value is 95.12% for after preprocessing. The difference between the recall value of before and after preprocessing is 0.91%. The results show that the accuracy of the automatic converted text after preprocessing is 0.91% improvement than before preprocessing. The false rate is 3.74% for before preprocessing and the false rate is 2.80% for after preprocessing. The difference between the false rate value of before and after preprocessing value is 0.94%. The results show that the false rate of the automatic converted text after preprocessing has decreased by 0.94% over the automatic converted text before preprocessing.

6. CONCLUSION AND FUTURE WORK

This paper works around the video to text conversion using Google Docs for the purpose of content analysis. Using Google Docs app, the superimposed audio is tracked from the video to convert it into text. The results of this work demonstrate that the various dimensions of the quantitative evaluation measurements are significantly differing case to case. The error rate is reduced through the preprocessing of the converted text. The error rate is reduced to a tune of 1% through preprocessing of the converted text. Hence, the preprocessed text can be used for text content analysis.

The limitation of this work is that the textual content covers only audio stream without considering the video stream part of it. Hence, the further work would consider some visual part of the video content so as to include the effects of visual gestures and postures and activities. In addition, the preprocessed text with noise and errors may be preprocessed again with different technique(s) if needed.

REFERENCES

Alper Kursat Uysal, & Gunal., S. (2014). The impact of preprocessing on text classification. Science Direct, Information Processing and Management, 104-112.

Ankur. (2016). Classification of Multi-Media Content(Video's on YouTube) Using Tags and Focal.

Bassem. (2013). Content-Based Video Browsing by Text Region Localization and Classification. International Journal of Video and Image Processing and Network Security IJVIPNS-IJENS, 10 (1).

Dongsong Zhang, J. F., & Nunamaker. (2004). A Natural Language Approach to Content-Based. IEEE Transactions on Multimedia, 6 (3).

Feinerer., I. (2015). Introduction to the tm Package Text Mining in R.

H. Bhaumik, (2016). Hybrid soft computing approches to content based video retrieval: A brief review. Appl. Soft Computing. J. Elsevier B.V., 1568-4946.

Kaladevi A.C & S.Theetchenya. (2013). Ontology Extraction for E-Learning - A Fuzzy Based Approach. IEEE Xplore .

Li, Y. (2006). Techniques for Movie Content Analysis and Skimming. IEEE Signal Processing Magazine.

M. Lyu, J. S., & Cai., M. (2005). A Comprehensive Method for Multilingual Video Text Detection, Localization and Extraction. IEEE Transactions on Circuits and Systems for Video Technology, 15 (2), 243-255.

Mehmood, I. (2016). Divide-and- Conquer based Summarization Framework for Extracting Affective Video Content. Neurocomputing,.

Mohammed Al-Zoube & ., B. K. (2010). A Data Mashup for Dynamic Composition of Adaptive Courses. The International Arab Journal of Information Technology, 7 (2).

Muenchen., R. (2013). The Popularity of Data Analysis Software. Technical report.

Parlakkilic & Karslioglu. (2013). Rapid multimedia content development in medical education. Journal of Contemporary Medical Education, J Contemp Med Edu , 1 (4), 245-251.

Pei-Chen Sun & Cheng., H. K. (2007). The design of instructional multimedia in e-learning: A Media Richness Theory-based approach. Science Direct Computers and Education, 49, 662-676.

S. Antani, R. K., & Jain., R. (2002). A survey on the use of pattern recognition methods for abstraction, indexing and retrieval of images and video, Pattern Recognition. 35, 945-965.

S. Vijayarani, M. J., & Nithya., M. ((2016).). Preprocessing Techniques for Text Mining - An Overview. International Journal of Computer Science and Communication Networks, 5, 7-16.

Vairaprakash Gurusamy, & Kannan., S. (2014). Preprocessing Techniques for Text Mining.

Williams., G. (2016). Hands-On Data Science with R Text Mining.

X. Hua, P. Y., & Zhang., H. (2002). Efficient video text recognition using multiple Frame Integration. IEEE Int. Conf. on Image Processing (ICIP) .

X.S. Hua, L. W., & Zhang., H. (2004). An Automatic Performance Evaluation Protocol for Video Text Detection Algorithms. IEEE Transactions on Circuits and Systems for Video Technology, 14 (4), 498-507.