iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 **Issue:** III **Month of publication:** March 2022

DOI: https://doi.org/10.22214/ijraset.2022.40775

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue III Mar 2022- Available at www.ijraset.com

Extractive Text and Video Summarization using TF-IDF Algorithm

Ajinkya Gothankar[1], Lavish Gupta[2], Niharika Bisht[3], Samiksha Nehe[4], Prof. Monali Bansode[5]
[1, 2, 3, 4, 5] *International Institute of Information Technology, Pune, India*

Abstract: ***Text summarization is a technique for extracting concise summaries from a large text without sacrificing any important information. It's a good way to extract crucial information from documents. The rapid rise of the internet has resulted in a substantial surge in data all across the world. It has become difficult for humans to manually summarise big documents. Automatic Text Summarization is an NLP technique that lowers the time and efforts required by a human to create a summary. Text summarising techniques are divided into two categories: extractive and abstractive. In the extractive approach, text summarising techniques choose sentences from documents based on a set of criteria. In the abstractive approach, text summarising techniques strive to improve sentence coherence by reducing redundancies and explaining the context of sentences. The extractive summarization approach is the subject of this paper. There are several methods for summarising data, including TF-IDF, Text Rank, PageRank, and Latent Dirichlet Allocation (LDA). This work examines Text Summarization using the TF-IDF Algorithm, a numerical measure that ranks the value of a word in a document based on how frequently it appears in that document and a set of documents. The application of the TF-IDF Algorithm for text, document, article, and video summarization is described in this study. There are no repetitions in the results, and for some searches, they are nearly identical to the summary results provided by humans. This algorithm offers a sentence extraction technique that selects the most diverse top-ranked sentences.***

Keywords: Extractive Summarization, Term Frequency-Inverse Document Frequency (TF-IDF), Natural Language Processing (NLP), Text Summarization, Video Summarization.

I. INTRODUCTION

Nowadays, the amount of data on the web is increasing exponentially on any topic. The rapid expansion of the internet resulted in a tremendous rise in the amount of information available, particularly in the area of text documents (e.g., news articles, e-books, scientific papers, blogs, tweets, etc.). Due to the massive amount of data circulating in the digital environment, much of which is unstructured textual data, separating meaningful information from the massive quantity of documents has become unfeasible. As a result, it is needed to automate solutions for understanding, indexing, classifying, and presenting all the information clearly and succinctly, allowing users to save time and resources.

Text summarization can be extractive or abstractive, depending on the method used to create the summary. Concatenating significant sentences extracted from the document to be summarised produces an extractive summary. An abstractive summary, on the other hand, communicates the key information from the texts. Abstractive summarization necessitates a significant amount of natural language processing. As a result, it's more difficult than extractive summarization. Extractive summarization has become a standard in document summarization due to its better achievability. This type of summarization uses statistical methods including the title method, location method, Term Frequency-Inverse Document Frequency (TF-IDF) method, and word method to extract key phrases or keywords from a document.

The Term Frequency-Inverse Document Frequency (TF-IDF) is a numerical statistic that indicates how essential a term is to a document in a corpus or collection. In information retrieval and text mining, this strategy is frequently employed as a weighting factor. TF-IDF is mostly used in text summarising and categorization applications to prevent words from being filtered out. The TF-IDF value rises in proportion to the number of times a word appears in a document and is offset by the word's frequency in the corpus, which helps to regulate the fact that some words are more common than others. The raw frequency of a phrase in a document is referred to as the term frequency. Furthermore, inverse document frequency is a metric for determining if a phrase is common or uncommon across all documents, which is calculated by dividing the total number of documents by the number of documents containing the term. In this paper, an extractive text summarization method called TF-IDF is used to build the summary.

II. LITERATURE REVIEW

While making this project we reviewed many research papers, these papers conveyed the latest research in this field, a summary of a few of the papers we referred to are mentioned below:

A. *NLP Based Machine Learning Approaches for Text Summarization*

In this paper, we have learned the use of various algorithms and methods. These methods, either individually or combined give different types of summaries. Their accuracy can be compared to find a better and more concise summary. For this purpose, the ROGUE score has been used more frequently. Similarly, in some cases, TF-IDF scores have been used too.

B. *Comparative Assessment of Extractive Summarization: TextRank, TF-IDF and LDA*

In this paper, the author has analysed and compared the performances of three different algorithms. Initially, different text summarization techniques were explained. The paper focused on the extractive approach. For comparison, three extraction algorithms namely TextRank, TF-IDF, Latent Dirichlet Allocation (LDA) were used. ROUGE 1 is used to evaluate the effectiveness of the extracted keywords. The results of the algorithms were compared with each other and also with the handwritten summaries, hence the performance was evaluated.

C. *Video Summarization using NLP*

This research offers an autonomous video summarising system based on natural language processing (NLP). This paper aims to produce a concise video summary that summarizes various YouTube/ Social Media videos. The suggested method generates a summarised video by first summarising the YouTube video transcripts. A web application is also being created that accepts input from the user in the form of a YouTube movie link and the desired summary duration.

III. PROJECT STATEMENT

Text summarization is a technique for condensing extensive passages of text. The goal is to develop a logical and fluent summary that only includes the document's major ideas. With the proliferation of digital media and ever-increasing publishing, who has the time to read complete articles, documents, or books to determine whether or not they are useful?

The following are some of the objectives for text summarization

1) To keep up with world affairs by listening to the news.
2) Investors make selections based on stock market updates.
3) People even go to the movies based on what they've read in the reviews.

People can make more effective decisions in less time with summaries. The goal is to construct a tool that is computationally efficient and automatically generates summaries.

We propose in this project to develop an extractive-based summarising method that employs TF-IDF (Term Frequency - Inverse Document Frequency) and POS (Parts of Speech) tagging to provide a thorough summary. We also propose to apply the same algorithm to evaluate the summary for videos.

IV. SOFTWARE AND HARDWARE

1) Coding Language: Python
2) Frontend: HTML, CSS, JavaScript, React.js
3) Backend: Python, Flask
4) Development IDE: Visual Studio Code
5) Server: Node Server (Frontend), WSGI Server (Backend)
6) CPU: 2.9Ghz (C2D)
7) RAM: 1Gb
8) HDD: 128Gb
9) Motherboard: Intel 945 GLX

V. EXISTING SYSTEM

Current summarization algorithm implementations are stiff and have input format restrictions, they also have limited control over the summarization output size, unlike our project implementation. We also have added a unique feature to our project which is Video Summarization. Additionally, the GUI of our project is extremely intuitive and user-friendly.

VI. METHODOLOGY

A. *Technique*

The following are the two techniques of text summarization:

1) *Extraction-based Summarization:* In Extractive Summarization, the most essential sentences from the total text data are selected and listed together as a summary.
2) *Abstraction-based Summarization:* In Abstractive Summarization, the summarizer first grasps the document's core concepts before generating new sentences that aren't found in the original.

In this paper we will be focusing on Extraction Based Summary:

Extractive summaries are created by extracting key text segments from the text (sentences or passages). The "most important" content is considered the "most frequent." As a result, no effort is expended on deep text comprehension. They are simple in concept and easy to implement.

B. *Algorithm*

Term frequency - Inverse document frequency (TF-IDF)

TF-IDF stands for term frequency–inverse document frequency and is a numerical metric used to rate the importance of a word in a document based on how often it appears in that document and a set of documents. The idea behind this metric is that if a term appears frequently in a document, it must be significant, so we should assign it a high score. However, if a word appears in too many other texts, it is likely not a unique identifier, and we should give it a lower score.

The formula for calculating TF and IDF:

1) TF(w) = (Number of times term w appears in a sentence) / (Total number of terms in the sentence)
2) IDF(w) = log10(Total number of sentences / Number of sentences with term w in it)

Hence TF-IDF for a word can be calculated as: TF-IDF(w) = TF(w) * IDF(w)

VII. FEATURES

A. *Text Summarization*

Purpose	Convert input text of various formats to summarized text using summarization algorithm
Input	1. Raw Text 2. Article 3. Document
Method	Extracting the text- Raw Text: It can be directly extracted using python inbuilt file reading operation. Article: We used Newspaper library of python for extracting the text from articles. Document: 1. PDF: We used PDFPlumber for extracting the text from a pdf file. 2. DOCX: We used Docx2txt for extracting the text from a docx file. 3. TXT: It can be directly extracted using python inbuilt file reading operation. Summarizing the extracted text using TF-IDF Summarization Algorithm.
Output	Successfully summarized the input text.
Application	Shortens reading time, speeds up research and expands the quantity of information that can be stored in a given space.

B. Video Summarization

Purpose	Summarize the video with the help of generated transcript.
Input	MP4 video file
Method	1. Upload the video. 2. Transcript of the video will be generated using google cloud speech API. 3. Summarized text will be obtained from the generated transcript of the video.
Output	Successfully summarized the video.
Application	Shortens reading time, speeds up research and expands the quantity of information that can be stored in a given space.

VIII. LIBRARIES

A. Flask

1) Flask is a lightweight framework that gives abundant features without external libraries and has minimalist features.
2) Flask is a Python framework for building web applications.
3) There includes a built-in development server as well as a quick debugger.

B. NLTK

1) NLTK is a popular Python programming language for working with human language data.
2) It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, semantic reasoning, wrappers for industrial-strength NLP libraries and an active discussion forum.

C. React

1) React is a front-end JavaScript library for creating user interfaces using UI components that is free and open-source
2) Used to build single-page applications and to create reusable UI components.

D. CORS

1) Cross-Origin Resource Sharing (CORS) is an HTTP-header based mechanism that allows a server to indicate any origins (domain, scheme, or port) other than its own from which a browser should permit loading resources.
2) The same-origin security policy forbids cross-origin access to resources.

E. Newspaper

1) Newspaper is a Python module used for extracting and parsing newspaper articles.
2) It uses advanced algorithms with web scraping to extract all the useful text from a website.

F. PYDUB

1) To work with audio files, Python has a package called Pydub.
2) Pydub is a Python library that works exclusively with WAV files.
3) We can play, split, merge, and edit WAV audio files with this library.

G. FLASHTEXT

1) FlashText is a Python library created specifically for searching and replacing words in a document.
2) The keywords are then looked for or replaced in the string using FlashText.

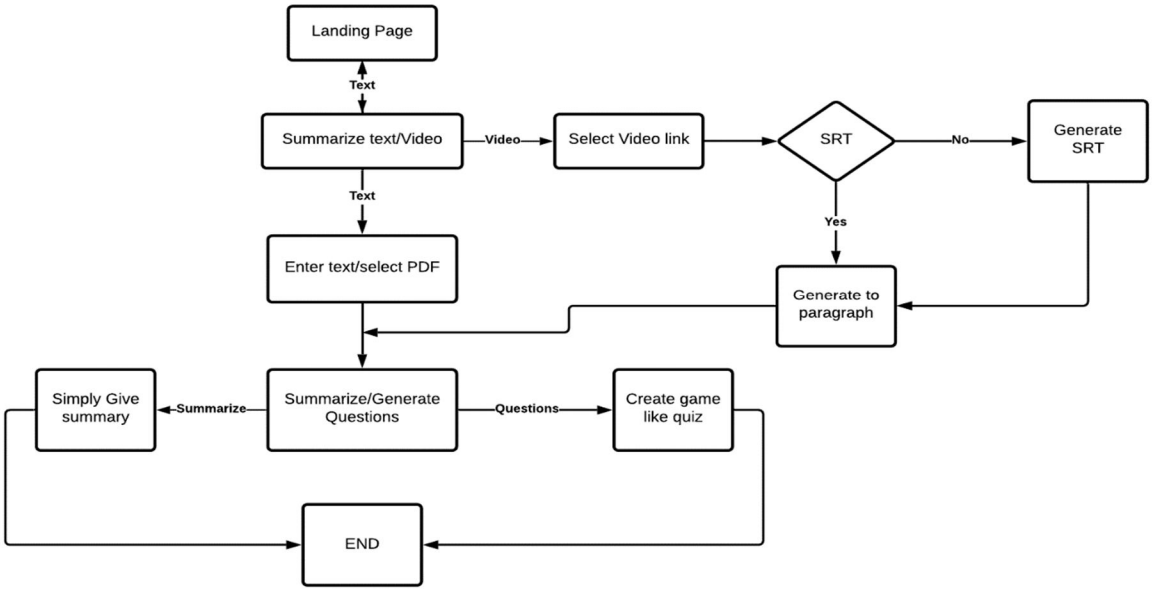
H. PDFPLUMBER

It obtain precise information on each text character, rectangle, and line by downloading a PDF.

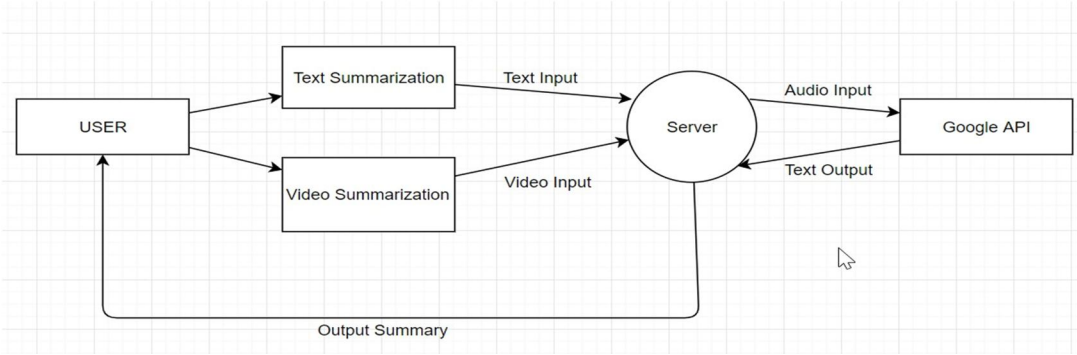
I. DOCX2TXT

It extracts higher-quality text by fixing common scan errors.

IX. SYSTEM DESIGN



X. DATAFLOW DIAGRAM



XI. IMPLEMENTATION

A. Taking User Input

First, we take the input from the user in various formats such as raw text, article, document, and video, along with the percentage of the summary. For this, we created a simple user interface in which the user can enter data in various forms and the summary will be generated by clicking the submit button.

B. Processing the Input

The input data is then converted to raw text in the backend using various libraries. For articles, we used the newspaper library, we used docx2txt and pdfplumber for the documents and we used the Google Cloud Speech API for the videos.

C. Pre-Processing and Summarization

Before summarizing, we eliminate stopwords and then lemmatize the words. After pre-processing, we summarise the text by calculating the TF-IDF score of each sentence and then selecting the number of top-scored sentences based on the ratio given by the user. We then reorder the sentences in the original format before displaying it.

D. Display Data

After summarization, the summarized content is delivered back to the frontend in JSON format, and the user can see it on our UI.

XII. RESULT

In this project we have successfully used TF-IDF to first summarize the text in multiple input formats, that is raw text, article (from a third-party website) and document (.txt, .docx, .pdf). Secondly, the same summarization algorithm was tweaked to generate a transcript from the video (.mp4) and then successfully generate a summary from the generated transcript.

XIII. FUTURE WORKS

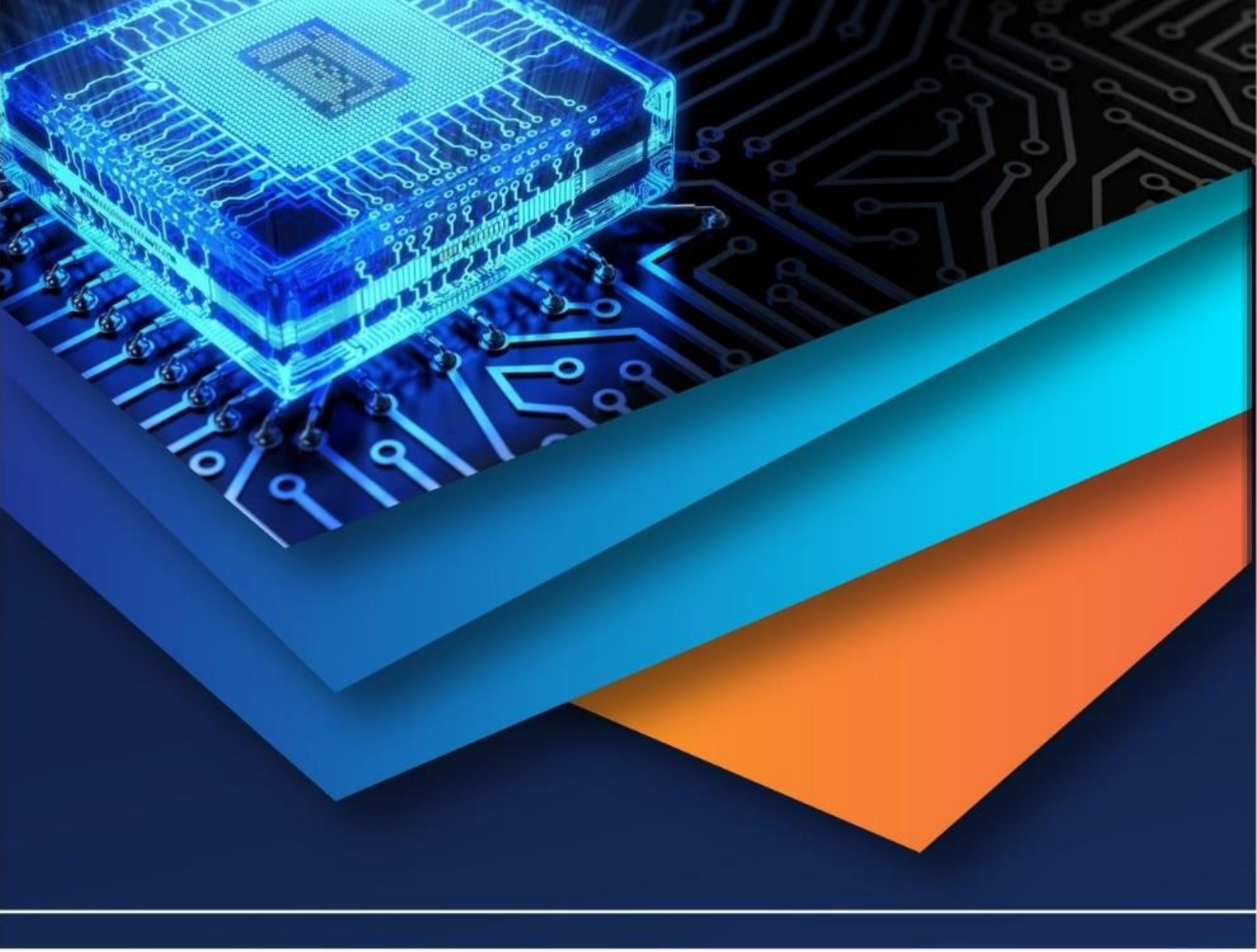
A. In the future, we are going to implement summarization for different languages such as Hindi, Marathi, etc.
B. We also look forward to adding voice/audio input support to the summarization algorithm.
C. We also have a goal of making our application available on as many devices and platforms as possible.

XIV. CONCLUSION

Technology to summarize is the need of the hour, hence we are creating an application to summarize text and video. Text summarization reduces reading time, speeds up the research, and increases the quantity of information that can be stored in a given space. Text summarising is a rapidly growing field, with specialized tools being created to handle more complicated summarization tasks. Users are expanding the use case of this technology, as open-source software and word embedding packages become more widely available. Automatic Text Summarization is useful for Natural Language Processing tasks like Question Answering, Text Classification, as well as other computer science domains like Information Retrieval, where the access time for information seeking will be improved.

REFERENCES

[1] NLP based Machine Learning Approaches for Text Summarization https://ieeexplore.ieee.org/abstract/document/9076358
[2] Single Document Automatic Text Summarization using Term Frequency-Inverse Document Frequency (TF-IDF) https://www.researchgate.net/publication/318963563_Single_Document_Automatic_Text_Summarization_using_Term_Frequency-Inverse_Document_Frequency_TF-IDF
[3] A Survey of Text Summarization Techniques https://link.springer.com/chapter/10.1007/978-1-4614-3223-4_3
[4] Automatic text summarization: A comprehensive survey https://www.sciencedirect.com/science/article/abs/pii/S0957417420305030
[5] Semantic Text Summarization of Long Videos https://www.researchgate.net/publication/316948434_Semantic_Text_Summarization_of_Long_Videos
[6] Automatic Multiple Choice Question Generation from Text: A Survey https://ieeexplore.ieee.org/document/8585151
[7] A Statistical Approach for Automatic Text Summarization by Extraction https://www.researchgate.net/publication/224250354_A_Statistical_Approach_for_Automatic_Text_Summarization_by_Extraction
[8] Assessing sentence scoring techniques for extractive text summarization https://booksc.eu/book/21495570/83518a
[9] Video Summarization using NLP https://www.irjet.net/archives/V8/i8/IRJET-V8I8411.pdf
[10] NLP based Machine Learning Approaches for Text Summarization https://ieeexplore.ieee.org/abstract/document/9076358
[11] VSUM: Summarizing from videos https://ieeexplore.ieee.org/abstract/document/1376676
[12] Automatic text summarization: A comprehensive survey https://www.sciencedirect.com/science/article/abs/pii/S0957417420305030
[13] Newspaper: Article scraping & curation https://www.geeksforgeeks.org/newspaper-article-scraping-curation-python
[14] Automatic Extractive Text Summarization using TF-IDF https://medium.com/voice-tech-podcast/automatic-extractive-text-summarization-using-tfidf-3fc9a7b26f5
[15] Reactstrap https://reactstrap.github.io/



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY