Available online to ejournal.pnc.ac.id

**Journal of Innovation Information Technology and Application**

Journal Page is available to https://ejournal.pnc.ac.id/index.php/jinita

# Autocomplete recommendation plugin and Summarizing Text using Natural Language Processing

*Aryaan Shaikh[1], Nikita Newalkar[2], Sakshi Gaikwad[3], Namrata Kadav[4], Chaitali Shewale[5]*

[1,2,3,4,5]*Department of Information Technology, Vishwakarma Institute of Information Technology, Pune, Maharashtra, India*

Email : [1]Aryaan.21910912@viit.ac.in, [2]Nikita.21910942@viit.ac.in, [3]Sakshi.21910962@viit.ac.in, [4]Namrata.21911024@viit.ac.in, [5]Chaitali.Shewale@viit.ac.in

**ARTICLE INFO**

**Article history:**
Received 24 May 2023
Revised 30 August 2023
Accepted 16 October 2023
Available online 30 December 2023

**Keywords:**
Autocomplete Abbreviation,
Microsoft Office Add-in,
Natural Language Processing,
Text Rank,
Text summarization.

**IEEE style in citing this article:**
A. Shaikh, N. Newalkar, S. Gaikwad, N. Kadav, and C. Shewale, "Autocomplete recommendation plugin and Summarizing Text using Natural Language Processing," Journal of Innovation Information Technology and Application (JINITA), vol. 5, no. 2, pp. 116–123, Dec. 2023.

**ABSTRACT**

Expert-caliber documents, reports, letters, and resumes can be easily developed using Microsoft Office. Microsoft Office offers capabilities such as grammar check, text and font checking & formatting, HTML compatibility, advanced page layout, image support, and more in contrast to a plain text editor, however, it does not have the autocomplete abbreviations feature. The paper proposes an Autocomplete abbreviation Recommendation System that will integrate the benefits of getting automatic suggestions of either full forms, abbreviations, or both by clicking on the option that is being suggested. This will provide more flexibility to the user using existing Microsoft Office platforms. To create this feature, we have examined the JavaScript JQuery functions to implement a basic autocomplete feature. Information overloading is also one of the most important problems brought on by the Internet's explosive expansion. Massive quantities of text are difficult for people to manually summarise. Thus, there is now a greater need for summarizers that are more sophisticated and potent. Hence, Python's packages, methods, and NLP are used in this work to implement Text Summarization. By using this technique, the phrase's overall meaning is enhanced and the reader's comprehension is enhanced.

## 1. INTRODUCTION

With the help of the latest technology and skilled environment, a plugin application can be developed for Microsoft application users. By using Web technology i.e., HTML, CSS, JavaScript, we are going to implement an abbreviation plugin and with the help of an NLP algorithm, we are going to make a text summarization web application. Loopholes amongst the papers or future work mentioned in other research are carried out for this research with gap identification and thus trying to resolve the same. To solve a real-world problem, this Web-based plugin can be used for other Microsoft applications by simply altering the code, such applications have been specified. Thus, an effort to solve a real-world problem with a lot of uncertainty in factors affecting it. Also, an exploration of various API-based functions and algorithms was done to solve the same.

People are becoming smarter and the world is moving toward modernity. These helpful add-ins/plugins for Microsoft apps, which may represent the end of the tunnel, are some of the best instances of modern technology. Today, technology has an incalculable impact on our lives. Generally, we can develop Microsoft add-ins with different problem statements in mind. These add-ins provide special customized functionality to these Microsoft applications as follows. Firstly, it will increase the capability of Office clients by integrating external data, automating documents, exposing Microsoft and third-party capabilities, and more. Use the Microsoft Graph API, for instance, to link to data that boosts productivity. Secondly, it



**) Corresponding Author : Aryaan.21910912@viit.ac.in*

will create new interactive, rich objects that may be included in Office documents. Add interactive visualizations such as maps, charts, and graphs that users can add to their own PowerPoint presentations and Excel spreadsheets.

The purpose of developing this "Microsoft Abbreviation Addin" is to help users write the full forms of Abbreviations OR Abbreviations of respective full forms in Microsoft Office platforms such as Microsoft Word, Microsoft PowerPoint, and Microsoft Excel. The user can accept a suggestion. This document describes the entire process which will be accomplished for building the project. Also, the purpose of developing "Text summarization" is that it cuts down on the time spent sifting through lengthy papers or articles by using fewer words to express key information. The acronym TLDR, or too long didn't read, is commonly used to describe summaries intended to make the material more easily understood. The amount of text material available online is constantly increasing, necessitating the use of automatic text summarization techniques to better assist in the discovery of pertinent information and speed up the consumption of pertinent information.

This abbreviation plugin will be a new feature that will be added to Microsoft Office. Through this abbreviation plugin, the users will now be able to write full forms or abbreviations of words without remembering them by getting suggestions. The main intent of building this plugin is to formalize the procedure of writing and remembering whole words and their abbreviations and make it easy for users. The plugin reports notable findings on user preference for abbreviation features and suggests best practices for their implementation. Text summarizer is a web application where users will be able to read a summary of long paragraphs which is a time-reducing technique. This web application consists of two fields Input text Field and Output text Field. Users have to put the long paragraph into the Input field to get the summary by clicking on the Button.

Similarly, we thought of creating this web add-in for not just Microsoft Word but also for the different Microsoft solutions like Microsoft Excel and Microsoft PowerPoint. Various Research Institutes and Agencies like the Defence Institute of Advanced Technology (DIAT) stated the problem faced by their documentation heads in memorizing these abbreviations and their correlated full form. Also, it is very difficult sometimes to search through large documents with written abbreviations which becomes a very tedious task to do. So, we came up with an idea to resolve this issue using a Microsoft plugin which is similar to a web solution for the documentation. The Microsoft autocomplete abbreviation plugin helps end users search for particular short forms related to a full form and vice versa. It gave additional ability to the add-in to add that into the content of the Word document. For this requirement, we thought of developing a single web page application that will summarize the text for the end user and will reduce his/her efforts to go through the whole document and give proper, meaningful insights from a summary in a short interval of time. Thus, ultimately helping to invest less time in this work.

## 2. LITERATURE SURVEY

A huge source of electronic information is the Internet. But the process of gathering knowledge ends up being a tiresome effort for people. As a result, automated summaries started to use our limited time to look for ways to automatically retrieve data from documents. The first automatic summary of the text was created in 1958 by H.P. Luhn Boo, Patricia Anthony, and Vooi Keong, [1], a data structure is shown that can be used to construct an autocomplete service in a typical computer that can search through millions of concepts. The proposed data structure reduces memory consumption while escalating search complexity. A similar lookup service to DBpedia that has been built and contains a words gallery of 9 million as completion candidates allows for the evaluation of the effectiveness of this data structure. Results show that this particular data structure requires lower memory than a ternary search tree and, more importantly, is capable of doing a lookup in a couple of milliseconds.

Fung, and Wong Jiang [2] described a technology called predictive text completion that goes beyond the conventional autocompletion and text replacement methods. It helps to cut down on the number of keystrokes needed for text entry in comparison to more expensive speech-to-text technology or specialized input devices, and it serves as a low-cost assistive technology for computer users with learning/reading impairments, dyslexia, or other disabilities. Python is used for quick R&D and testing of new features, whereas AutoHotKey can be used to create the standard stable version.

This research [3] presents a hybrid approach that makes use of the convergence between machine learning methods and sophisticated design methodologies to construct a code auto-completion system that allows firmware designers to write qcode more successfully. The probabilistic design provides a list of possible next software components that may be individually selected to increase prediction accuracy,

whereas the deterministic design diminishes prediction accuracy because it delivers output in unexpected ways.

For the incomplete model, NLP-based assistance will offer autocomplete suggestions that are being built for the project depending on the project's available textual data's automatic analysis (contextual knowledge) and its connected business domain, as suggested by Burgueño et al. [4]. This will simplify domain model definition and improve the caliber of such models. The designer's responses to the assistant's feedback will also be considered during the process. They created a proof-of-concept program, and our preliminary study yielded encouraging results.

Sarker, Soumik, Jillur Rahman Saurav, Md Mahadi Hasan Nahid, and Md Ekramul Islam [5], suggested a trie, sequential LSTM, and N-gram integrated methodology for word completion and sequence prediction in Bangla. To store Bangla vocabulary and get the term from a user-input prefix, the trie data structure was implemented. The authors looked into a mixed neural network and N-gram technique for sequence prediction. Better performance than any single model implementation is revealed by this sequential LSTM and N-gram cooperation. For more effectiveness, the model was tested using both small and large-scale Bangla datasets. The results of the studies indicate that our hybrid strategy for word completion and sequence prediction offers promise. Our architecture, in our opinion, has a significant influence on Bangla search engines, keyboards, and future recommendation system-based research. The authors Allahyari, Mehdi, et al. [6] have highlighted different extraction methods for summarizing single and many documents. The most popular procedures, including graph-based practices, domain depiction techniques, machine learning techniques, and frequency-driven approaches, have all been demonstrated. The authors believe that this study gives a good summary of present developments and patterns in automatic comprehensive analysis methods and explains the condition in this research area, but besides the fact that it's not possible to discuss all conceivable methods and algorithms in this work.

By contrasting the accurateness of automatic performance measures with the highest-scoring output results, both extractive and abstractive, using the most popular datasets for both root-level and summary-level assessment settings, Manik, Pranav Gour, et al. [7] have aimed to re-evaluate the evaluation process for text summarising. They discovered that generalizations about evaluation measures drawn from more recent datasets and systems are not always true. They made available a dataset of human evaluations compiled from the top 25 neural summarization algorithms (14 abstractive and 11 extractive). Their research not only identifies the shortcomings of the existing metrics but also emphasizes the necessity of updating the meta-evaluation testbed to keep up with the quick development of new systems and datasets.

The COMPENDIUM text summary system was assessed by María Teresa Romá-Ferri, Lloret, Elena, and Manuel Palomar [8] for its appropriateness in creating abstracts for biological publications. Two approaches are proposed: one is COMPENDIUME (extractive), which simply chooses and abstracts the most important lines from the texts, and the other is COMPENDIUME-A (abstractive-oriented), which additionally handles the issue of abstractive summarization. They aimed to learn more about the best summary approach, how good COMPENDIUM is at producing summaries for the biomedical field, and how actual users see automatic summaries. As a result, the substance of the summarized text as well as the degree of user approval were evaluated using two distinct evaluation methods: quantitative and qualitative. The findings show that the performance of the information in abstractive-oriented and extractive summaries is comparable. This means that while both systems are capable of maintaining the pertinent info when evaluating user happiness, the latter technique is superior from a human perspective because it includes information from the source materials.

The authors of the paper [9] have offered a thorough and organized analysis of the text summarizing research that has been done from 2008 to 2019. The results of the analysis provide a thorough description of the subjects and patterns that have been the focus of research in the area of text summarization. They also detail the many approaches and procedures that researchers typically employ for method creation and comparison, as well as references to pre-processing, features, and open datasets that have been utilized. Several suggestions for improvements and issues with text summarizing research are made after this publication.

El-Kassas et al. [10] have presented the various ATS components; methods, procedures, building blocks, approaches, datasets, assessment techniques, and forthcoming study directions. They have completed a thorough survey for the researchers. The main objective of an ATS system is to produce a summary that distills the essential ideas from the input information while avoiding duplication. Without having to read the complete document, the ATS systems support workforces in understanding the major aspects of the original document. The automatically generated summaries can help users and save their time and work. Gambhir, Mahak, and Vishal Gupta [11] have provided a thorough analysis of the most recent

extractive methods of text summarization created in the previous 10 years. Their requirements are noted, and comparative advantages and disadvantages are listed. There are also several abstractive and multilingual text summarizing techniques mentioned. In this area of research, summary evaluation is still another difficult problem. As a result, conferences and seminars on text summarization evaluation as well as intrinsic and extrinsic summary evaluation methodologies are detailed. Additionally, evaluation findings for extractive summarization techniques are shown using some common DUC datasets. Finally, this essay comes to a close with a discussion of helpful future approaches that can aid researchers in determining areas in which additional study is required.

The analysis's findings [12] give a detailed explanation of the issues and trends that are the subject of their text summarization research; they also provide references to open datasets, pre-processing, and features that were employed, and they describe the approaches and techniques that researchers frequently use for method development and comparison. There are various suggestions made regarding opportunities and difficulties in the field of text summarising research which serves to choose an efficient and sustainable method for text summarization.

In the study[13], measurements and complicated network ideas were used to choose phrases for an extracting summary. The edges connecting sentences with similar significant nouns form the nodes of the graphs or networks that represent a single piece of text. Since the metrics of such networks are thought to be able to capture key text properties, using complex networks to express texts seems to be appropriate for automatic summarization. Via usability studies on its application in common academic research contexts, this paper[14] investigates a technique that utilizes auto-suggest in search interfaces. This study offers best practices for their implementation as well as noteworthy insights on client requirements for auto-suggest functionality.

Dima, A., & Massey, A. [15] contrasted two methods for extracting key phrases; firstly, employing a toolkit-based approach that considered word and phrase distributional characteristics, and the Maui automated concept indexer, a well-known research technique. The findings of this study indicated that when author-provided key phrases had first been eliminated from the text, the toolkit approach was competing with Maui. Gupta, V., & Lehal, G. S. [16] conducted research work on the problem of significant variance in retrieval techniques or document representations that affect how their combined outputs are combined. To address this problem, they suggested an original class-based data fusion method. The segments that are obtained by models utilizing various document representations are divided into three classes based on segment quality: low, intermediate, and high class. According to experimental findings, the novel technique used by them greatly outperformed CombSUM or WCombSUM in combining data with high-quality variations.

## 3. METHODOLOGY AND TOOLS USED

### 3.1. Recommendation Office Plug-in

A web application created by a web server or web hosting service (like Microsoft Azure to host the online application) and an XML manifest file are the two fundamental parts of an Office Add-in. The parameters for the add-integration with Office clients are defined in the manifest, among other things.

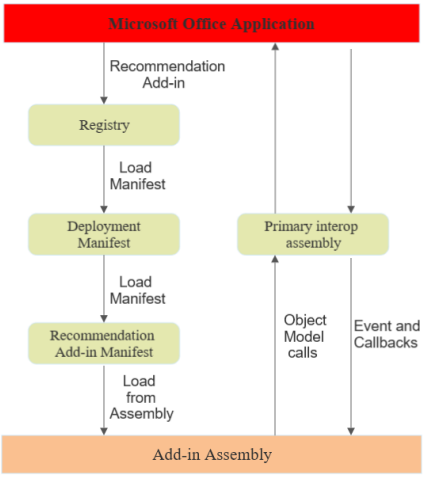


Figure 1. Flow for Microsoft Office Add-in

In Fig. 1 document is first sent for pre-processing, the text is cleaned up and organized during text pre-processing so that it can be fed into a model for additional analysis and learning. Text pre-processing methods might be generic, making them usable in several different contexts/applications, or they can be focused on a particular purpose. Spelling corrections, tokenization, stemming, and sentence segmentation are some stages involved in it.

As it is a web web-based add-in, we have implemented it as a web-based application and used HyperText Markup Language (HTML) for designing UI and CSS (Cascading Style Sheets) for decorating the UI and JavaScript to add functionalities to the web page add-in. Specifically, we have used the autocomplete function from jQuery API to create a dynamic dropdown menu of abbreviations and their respective full forms. Office Add-ins can virtually perform any task that a website can in a browser. The platform for Office Add-ins is utilized to increase the capability of Office clients by integrating external data, automating documents, exposing Microsoft and third-party capabilities, and more.

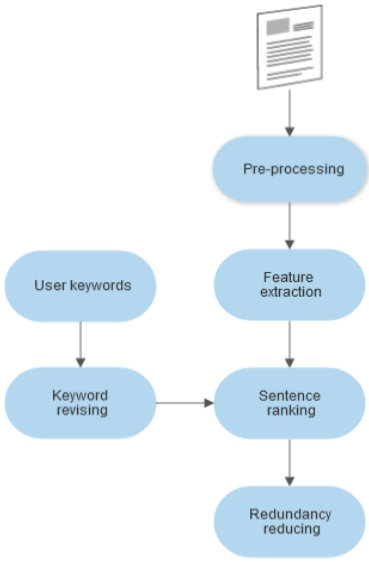


Figure 2. System Architecture of Text Summarizer

### 3.2. Text Summarization using NLP

The process of condensing a written document while retaining its core ideas and general message is known as text summarization. It is typically an issue in NLP and can be helpful for several applications, including automated question-answering systems, machine translation, and information retrieval.

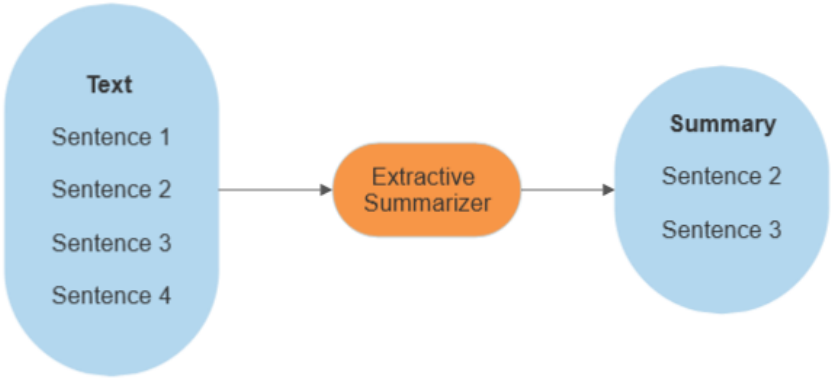


Figure 3. Extractive Summarizer

### 3.2.1. Extractive Summarization

We have utilized an extractive summarization technique using the Spacy NLP library using the test rank algorithm. Creating a summary of a larger text by choosing and pulling out the utmost crucial info from the original text is known as extractive summarization and is a sort of natural language processing (NLP) work. The original content should be condensed into this summary while keeping the most crucial details and highlighting its primary ideas.In sentence-based extractive summarization, the sentences in a text are ranked by importance, and the most significant sentences are chosen to be comprised in the summarized document. The sentences of a text can be ranked variously: Frequency-based, Position-based, Similarity-based, and Machine learning-based. For this research, we have preferred using Frequency-based, this method ranks sentences based on how often certain terms are used in them. The theory behind this is that the most significant words will appear more frequently in the text because they will be found in the most significant phrases. Using the frequency-based technique, firstly, sentences are ordered in the text according to their significance before creating a summary using a sentence-based extractive summarization algorithm. The best-ranked sentences would then be chosen to be included in the summary. To generate the summary, these sentences are integrated while taking care to maintain the consistency and sense of the original text.

### 3.3. Text Summarization Components and Steps

We have implemented a text summarizer with the help of the TextRank algorithm. By identifying the key phrases or sentences in a manuscript, an algorithm called TextRank can be used to summarize it. Each sentence or phrase is represented as a node in a graph that is constructed from the text to function. A ranking algorithm is then used to determine which words or phrases are the most crucial in the document by weighting the edges between nodes according to how similar the sentences or phrases are to one another.

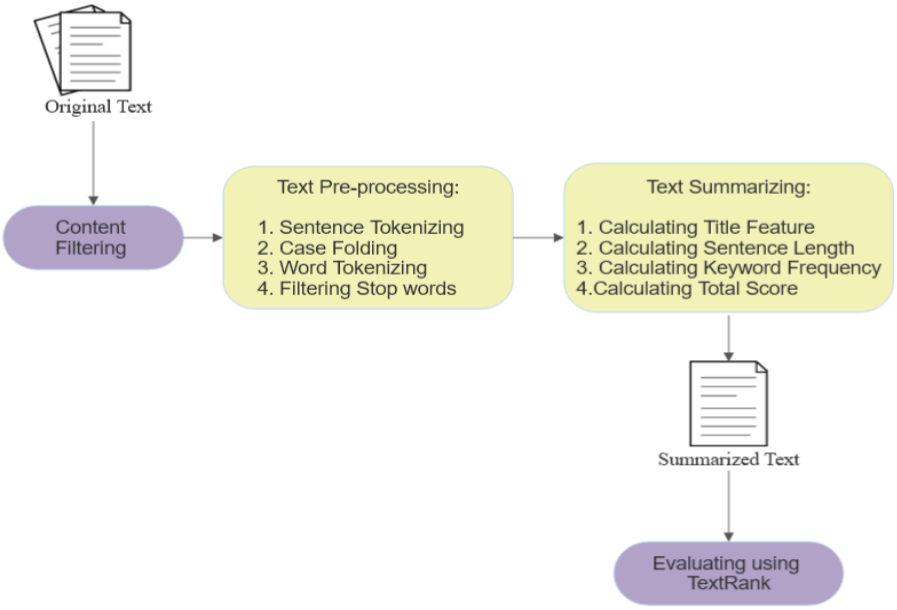


Figure 4. Text Summarizer Methodology

Fig. 4 involves the steps used to construct a text summarizer using TextRank. Firstly, the text is pre-processed by being tokenized into sentences, and then each sentence is broken down into words. Then created a text graph with each sentence or phrase acting as a node. The weights of the edges between nodes can be calculated using a similarity metric, such as cosine similarity. Orderly, it finds the most crucial phrases or sentences in the text and uses a ranking system like PageRank. For generating the text summary, this algorithm picks the strongest phrases or sentences. After generating the text summary, the summarized text is evaluated using the Text Rank algorithm.

### 3.4. Spacy

Natural language processing (NLP) in Python is made simple and effective using SpaCy. It is made to make it easier for programmers to create tools that effectively process and comprehend massive amounts of text. The method of automatically producing a condensed version of a text document while

retaining its most crucial information is known as extractive summarization, and it is one of the many valuable jobs that SpaCy can carry out. Installing the library and downloading one of its pre-trained language models are prerequisites for using SpaCy for extractive summarization. English, Spanish, and German are just a few of the pre-trained models that SpaCy provides. The summarizing feature of SpaCy can be used once the library and a language model have been set up.

Extraction of the key sentences from a text is one technique to use SpaCy to summarize it. Utilizing SpaCy's built-in summarizing tool, you may accomplish this. This feature assesses the significance of each sentence in the text based on attributes including length, placement in the document, and the presence of named entities. The top N sentences from the rated list can then be chosen to create your summary. Another approach for summarizing text using SpaCy can enable us to extract primary themes and create a summary based on those themes. Use SpaCy's topic modeling and named entity recognition (NER) features to do this. Topic modeling can assist you in identifying the primary themes or topics covered in the document, whereas NER enables you to extract significant named entities from the text, such as important persons, organizations, and locations. Then, using this data, you may create a summary that concentrates on the text's key themes and concepts. Thus, SpaCy is an effective and versatile tool for extractive summarizing, including a variety of functions and pre-trained models that can assist you in producing text document summaries rapidly and accurately.

### 3.5 Heapq module

The Python heapq module, which offers an implementation of the heap queue algorithm, contains the nlargest function. To locate the n largest elements in an iterable, use the nlargest function. The largest element lies at the very end of the list produced by nlargest because it returns the elements in ascending order. Although nlargest can help locate the list's biggest items, it is not frequently used for text summaries. Based on the content and structure of the text, a text summary often entails choosing a subset of the most crucial lines or phrases from a broader material. Text summarization can be accomplished using a variety of strategies, such as extractive and abstractive methods.

### 3.6. TextRank

A popular approach for summarizing text using NLP is the use of the TextRank Algorithm. It is predicated on the notion that the maximum crucial info in a document is frequently stated in a small number of key lines, and that these words can be found and chosen for inclusion in a summary by ranking them according to the quantity of new information they include. The algorithm parses the input text before using TextRank to identify the key terms and entities that are mentioned in the text. Then, based on how much fresh information each sentence in the document adds to the primary concepts and entities, it uses these concepts and entities to discover the relevance of each sentence. All sentences of the text are assessed according to their importance, and the ones with the highest positions are chosen to be contained within the summarized text. The TextRank algorithm's versatility in handling various input text kinds and languages is one of its main features. It can be trained to produce summaries that are targeted to particular sorts of documents or audiences, and it can be used to summarize materials written in multiple languages and from diverse disciplines. Overall, TextRank is a potent and popular method for automatically summarizing material written in natural language. Due to its versatility in terms of input formats and languages as well as its efficacy and efficiency, it is a crucial tool for numerous applications.

## 4. RESULTS AND DISCUSSION

An autocomplete abbreviation plugin for Microsoft applications would provide a feature that suggests and completes abbreviations as the user will be typing words. This can save time and improve productivity by allowing users to enter abbreviations quickly and accurately. The results of using an autocomplete abbreviation plugin for Microsoft applications will likely depend on the specific implementation and the user's needs and preferences. As a result, such a plugin could provide a range of benefits, including faster text entry, improved accuracy, customization options, and compatibility with other Microsoft applications. Figure 5. displays a brief view of the Automatic abbreviation recommendation plugin in different Microsoft Office platforms. The Autocomplete abbreviation Recommendation System will integrate the benefits of getting automatic suggestions of either full forms, abbreviations, or both by clicking on the option that is being suggested.

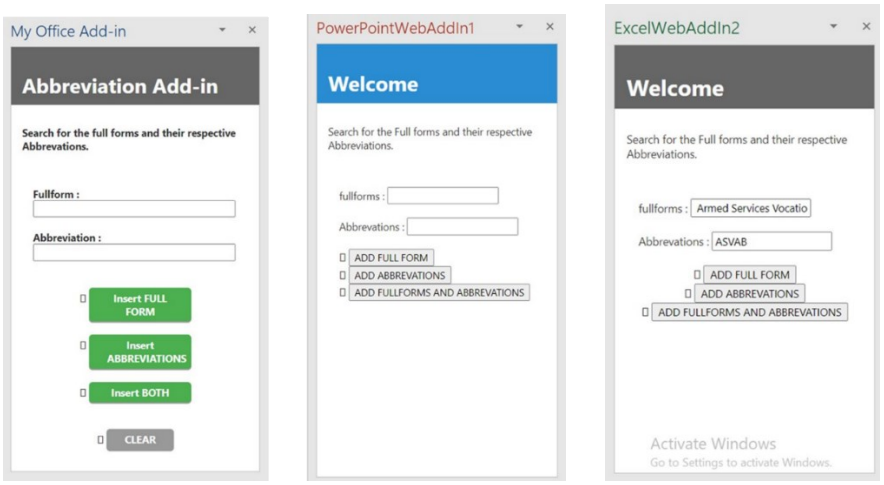


Figure 5. Automatic abbreviation recommendation plugin for different Microsoft Office platforms

Fig. 6 & Fig. 7 represent the insight of an overall view of how the plugin will predict and recommend words in Microsoft platforms like Word and Excel respectively. Fig. 6 shows options like Insert Full form, Insert Abbreviation, Insert Both, and Clear. These options can come in handy when dealing with a lot of abbreviations or words with complex full forms. Moreover, these options will help save time and make the platform easier to use.

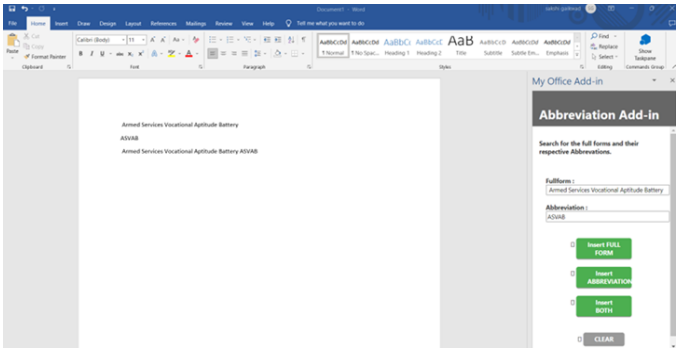


Figure 6. Automatic abbreviation recommendation plugin Microsoft Word Interface

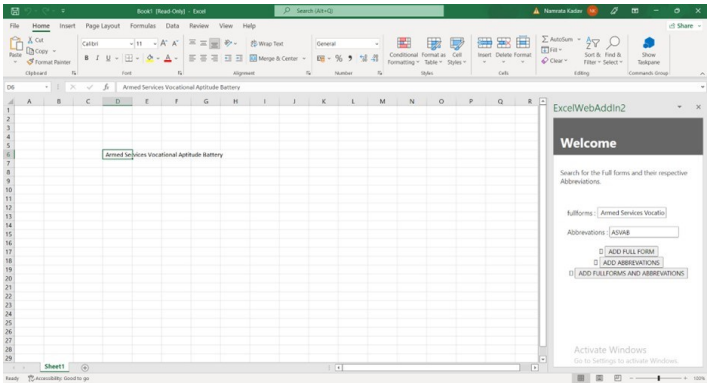


Figure 7. Automatic abbreviation recommendation plugin Microsoft Excel Interface

For the text summarizer algorithm, TextRank has produced a condensed version of the original text that only contains the most significant or pertinent passages or sentences. The summarizer's exact implementation and the settings you specify will determine how long it will be. For instance, you may set a fixed number of sentences or words for the summary, or you could decide to create a summary of a fixed proportion of the length of the original document text. Figure 8. displays that the summarized text is almost

33% of the original text. The effectiveness of the summarization algorithm and the suitability of the summary length for the given text and context have determined the better quality of the summary.

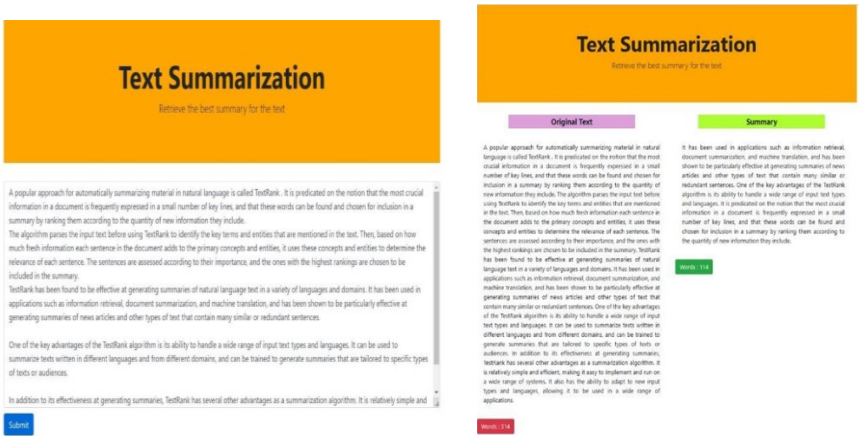


Figure 8. Text Summarization Original Text to Summarization

It is observed to be a good idea to evaluate the eminence of the summary by contrasting it with the original text and making sure that it accurately captures the main ideas and important details in the text. By doing so, we came to know that results are quite accurate and the web page is giving meaningful insights from the page. It's also important to keep in mind that text summarization is a demanding endeavor and that it might be difficult to produce effective summaries that faithfully reflect the meaning and content of the complete original text. The complexity of the resulting document, the existence of ambiguous or challenging-to-parse language, and the particular objectives or requirements of the summarization work are only a few of the numerous variables that might influence the quality of the summary. It is therefore typical to utilize several text summarizing algorithms and compare the outcomes to choose the optimal strategy for a particular situation.

## 5. CONCLUSION

NLP provides with the technology necessary to quickly extract the most important information from a given document. Our text summarizing algorithms are available and simple to use, improving the effectiveness and actionability of your research and corporate decision-making processes. The ability of NLP to extract information from text data makes it the ideal tool for tracking consumer feedback and identifying sentiment, including whether it is good or negative and to what extent. This research work can help eradicate the process of memorizing or referring to multiple abbreviation pages for writing documents. Text summaries are beneficial for problems in related fields of computer science, such as text categorization and information retrieval, as well as for tasks involving NLP, such as question-and-answer sessions. The speed of information search will also be increased. Moreover, the influence is also strengthened by sequencing, and algorithms are less biased than human creams. Commercial capture services give customers the ability to process more texts by using a text summary method. To conclude, the autocomplete abbreviation add-in for different Microsoft applications can be a useful tool for people working in diverse fields and areas, as it would allow them to quickly and easily type out full forms and abbreviations without having to memorize or reference multiple pages. The ability to add full forms and abbreviations to the list would allow users to customize the tool to their specific needs and workflows, and the overall goal of reducing efforts and improving efficiency would be achieved through the automatic completion of abbreviations as they are typed. This can be especially useful for people who frequently use technical terms or industry-specific language, as it allows them to type out long phrases with just a few keystrokes. As a result, businesses may track evaluations in real time, mark the most crucial or urgent comments, give fast feedback, and disregard unnecessary material.

## ACKNOWLEDGEMENTS

The authors would like to deeply thank of Defence Institute of Advanced Technology for providing this problem statement to work on and the faculty of the Information Technology Department of Vishwakarma Institute of Information Technology, Pune for their guidance, informative discussions, and plenty of suggestions for successfully carrying out this research study.

## REFERENCES


[1] V. K. Boo and P. Anthony, "A data structure between trie and list for auto-completion," in Knowledge Technology Week, pp. 303-312, Springer, Berlin, Heidelberg, 2011.
[2] W. J. Fung, "A predictive text completion software in Python," The Python Papers Monograph 2, 2010.
[3] J. Kim, K. Lee, and S. Choi, "Machine learning-based code auto-completion implementation for firmware developers," Applied Sciences, vol. 10, no. 23, p. 8520, 2020.
[4] L. Burgueño et al., "An NLP-based architecture for the autocompletion of partial domain models," in International Conference on Advanced Information Systems Engineering, pp. 91-106, Springer, Cham, 2021.
[5] S. Sarker et al., "Word completion and sequence prediction in Bangla language using trie and a hybrid approach of sequential LSTM and N-gram," in 2020 2nd International Conference on Advanced Information and Communication Technology (ICAICT), pp. 162-167, IEEE, 2020.
[6] M. Allahyari et al., "Text summarization techniques: a brief survey," arXiv preprint arXiv:1707.02268, 2017.
[7] M. Bhandari et al., "Re-evaluating evaluation in text summarization," arXiv preprint arXiv:2010.07100, 2020.
[8] E. Lloret, M. T. Romá-Ferri, and M. Palomar, "COMPENDIUM: A text summarization system for generating abstracts of research papers," Data & Knowledge Engineering, vol. 88, pp. 164-175, 2013.
[9] A. P. Widyassari et al., "Review of automatic text summarization techniques & methods," Journal of King Saud University-Computer and Information Sciences, 2020.
[10] W. S. El-Kassas et al., "Automatic text summarization: A comprehensive survey," Expert Systems with Applications, vol. 165, p. 113679, 2021.
[11] M. Gambhir and V. Gupta, "Recent automatic text summarization techniques: a survey," Artificial Intelligence Review, vol. 47, no. 1, pp. 1-66, 2017.
[12] A. P. Widyassari and S. R. (2020), "Review of automatic text summarization techniques & methods," Journal of King Saud University - Computer and Information Sciences, 18.
[13] L. Antiqueira et al., "A complex network approach to text summarization," Information Sciences, vol. 179, no. 5, pp. 584-599, 2009.
[14] D. Ward, J. Hahn, and K. Feist, "Autocomplete as research tool: A study on providing search suggestions," Information Technology and Libraries, vol. 31, no. 4, pp. 6-19, 2012.
[15] A. Dima and A. Massey, "Keyphrase Extraction for Technical Language Processing," UMBC Faculty Collection, 2021.
[16] V. Gupta and G. S. Lehal, "A survey of text summarization extractive techniques," Journal of emerging technologies in web intelligence, vol. 2, no. 3, pp. 258-268, 2010.