

**PROBEXPERT: AN ENHANCED Q&A PLATFORM
FOR REDUCING TIME SPENT ON LEARNING AND
FINDING ANSWERS**

2021-155

Ranasinghe R.M.A.K.

IT18011012

Bachelor of Science Special (Honors) in Information
Technology
Specializing in Software Engineering

Department of Computer Science & Software Engineering

Sri Lanka Institute of Information Technology
Sri Lanka

October 2021

**PROBEXPERT: AN ENHANCED Q&A PLATFORM
FOR REDUCING TIME SPENT ON LEARNING AND
FINDING ANSWERS**

2021-155

Ranasinghe R.M.A.K.

IT18011012

Dissertation submitted in partial fulfillment of the requirement for the Degree of
Bachelor of Science Special (honors) in Information Technology

Department of Computer Science & Software Engineering

Sri Lanka Institute of Information Technology

Sri Lanka

October 2021

DECLARATION

I declare that this is my own work, and this dissertation does not incorporate without acknowledgement any material previously submitted for a Degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text. Also, I hereby grant to Sri Lanka Institute of Information Technology, the nonexclusive right to reproduce and distribute my dissertation, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature: 

Date: 10/13/2021

The above candidate has carried out research for the bachelor's degree Dissertation under my supervision.

Signature of the supervisor:

Date:

ABSTRACT

In modern day Question and Answer platforms such as Quora, Stack Overflow has become a primary source of knowledge for those in the IT sector. However, the availability of vast amounts of information makes it considerably more difficult for people to solve their own problems. As a result, users are unable to solve their problems as they would want. We hope that this work will assist developers who wish to rapidly summarize the important elements of numerous response posts related to a technical query before reading the contents of the articles. For a given technical question, we define our work as a query-focused multi answer-post summarization assignment. This paper discuss how technologies can be used to reduce the time to get an optimized answer to the question by summarizing top voted answers. With this paper's main outcome, users will be able to find the exact solution they needed within seconds of time.

Keywords: answer optimization, text summarization, keyword extraction

ACKNOWLEDGEMENT

I would like to take the procession of this section to express my sincere gratitude to all the individuals who guided me through this journey since day one. First and foremost, I would like to appreciate the Sri Lanka Institute of Information Technology (SLIIT) for providing this chance to share new ideas through this project as a course requirement. Also, I take this opportunity to thank each faculty member and lecturer who lent their hand in guidance and support throughout this research project.

It was an honor to have a supervisor who assisted us to get back in the right direction when we chose the wrong path. So, I would like to express my heartiest gratitude to Ms. Dinuka Wijendra, who was willingly agreed to supervise this research through the year and provided advice to enhance the worth of end result. I would like to thank our co-supervisor Ms. Anjalie Gamage, who was willing to supervise this project throughout the year.

Finally, my gratitude would be paid to the colleagues of my team, my friends and my family members who encourage and support to strengthen.

Last but not least, I'd want to thank everyone else whose names aren't included here but who have shown their unwavering guidance and support in any way they can.

TABLE OF CONTENTS

DECLARATION	ii
ABSTRACT.....	iii
ACKNOWLEDGEMENT	iv
TABLE OF CONTENTS.....	v
LIST OF TABLES	vi
LIST OF FIGURES	vii
LIST OF ABBREVIATIONS.....	vii
1. INTRODUCTION.....	1
1.1 Background Literature.....	1
1.2 Research Gap.....	6
1.2.1 Automatic Text Summarization.....	6
1.2.2 Extractive approaches for Text Summarization.....	8
1.2.3 Abstractive approaches for Text Summarization	11
1.2.4 Question And Answer Quality	14
1.3 Research Problem.....	15
1.3.1 Research Question 1: Where to find the questions and answers?.....	16
1.3.2 Research Question 2: How to evaluate the effectiveness of an optimized answer?.....	16
1.3.3 Research Question 3: How to save time on use of an optimized answer?.....	17
1.4 Research Objectives.....	17
1.4.1 Main Objective.....	17
1.4.2 Specific Objective.....	18
2. METHODOLOGY.....	19
2.1 System Overview	19
2.1.1 Data Retrieval	19
2.1.2 Key word extraction.....	20
2.1.3 Check and remove similar answers	21
2.1.4 Text Preprocessing.....	22
2.1.5 Answer summarization	27
2.1.6 Answer merging	28
2.1.7 Check quality of optimized answer	29
2.2 Commercialization aspects of the product.....	29

Answer optimization option.....	30
2.3 Testing & implementation.....	31
2.3.1 Testing	31
2.3.2 Implementation.....	33
3. RESULTS & DISCUSSION.....	36
3.1 Results.....	36
3.2 Research Findings	37
3.3 Discussion.....	38
3.4 Summary of the Student Contribution.....	38
4. Conclusion.....	39

LIST OF TABLES

Table 1.1 - Feature comparison of Q-and-A platforms.....	15
Table 3.1 - Sample time taken for answer generation	37
Table 3.2- Student contribution.....	38

LIST OF FIGURES

Figure 1.1 - Summary of responses for what kind of resources people used to find the solutions.....	2
Figure 1.2 - Summary of responses for how much time spent on finding a solution for errors.	3
Figure 1.3 - Summary of responses for getting frustrated because available solutions did not solve the questions.	3
Figure 1.4 - Summary of responses for how many times users have to go back and forth in search of the correct answer.	4
Figure 1.5 - Summary of responses for finding the solutions they wanted in first try.	4
Figure 1.6 - Summary of responses for getting confused due to availability of multiple solutions.	5
Figure 1.7 -Automated text summarization approaches	7
Figure 1.8 - Extractive summarization.....	8
Figure 1.9 - Extractive summarization with TF-IDF.....	10
Figure 1.10 - Abstractive summarization	11
Figure 1.11- Abstractive summarization with LSTM.....	13
Figure 2.1 - System Diagram	19
Figure 2.2- Text preprocessing.....	22
Figure 2.3-Values given after Tokenization in a sample text.....	24
Figure 2.4- Stop words extracted from a sample text.....	26
Figure 2.5 - ProbExpert's Marketing Strategy.....	30
Figure 3.1- Sample answer extraction query	36
Figure 3.2- Output of the query.....	37

LIST OF ABBREVIATIONS

Q&A	Questions and Answers
NLTK	Natural Language Tool Kit
POS	Part-of-speech
BERT	Bidirectional Encoder Representations from Transformers
TF-IDF	Term Frequency–Inverse Document Frequency

1. INTRODUCTION

1.1 Background Literature

The scale and pace of information have increased considerably due to the rapid growth of the world wide web. Unfortunately, the abundance of knowledge has resulted in the problem of information overload[1]. It occurs when people struggle to find the proper information due to low information processing capabilities. To address such a problem, different information technologies such as search engines, recommender systems, question and answer systems[2] have been presented. These solutions strive to offer users with appropriate information as rapidly as possible based on their stated or implicit demands. Another sort of solution, such as information extraction, and text summarizing is to provide users with a simplified representation of information[3]. The purpose of this thesis is to assist lessen the amount of information that specially the software developers must deal with.

Answers on Q&A sites such as Stack Overflow[4] and Quora have become an important source of information for developers to address technical challenges in recent years[5]. The majority of developers will use a search engine to submit their requests, which will return a list of related postings that may contain answers. Then, in order to solve their problems, developers must look over the responses and gather knowledge. On the other side, the sheer volume of questions and answers on the aforementioned Q&A websites makes it difficult to find information. The majority of developers will use a search engine to submit their requests, which will return a list of related postings that may contain answers. Then, developers must go over the answers and gain knowledge in order to find solutions to their problems. As a result of such issues, it may take a long time for developers to receive the exact solution they require. It's also tough to discover answers in long articles because there's a lot of noise and duplicate content online, and the solution they find may only solve part of the problem.

A survey was carried out with the participation of 50 undergraduates at the Sri Lanka Institute of Information Technology using a survey questionnaire that included both fundamental and advanced questions that would be useful in this experiment. The primary purpose of the fundamental questions was to identify whether or not undergraduates were experiencing the same problems that the research would address. Based on the survey results and as shown in the figure 1.1, the undergraduates preferred to use Q&A sites to find solutions to challenges they encountered throughout their studies rather than books, documents, or tutorials. The key reason for this was that it saved time that would otherwise be spent manually going through other resources.

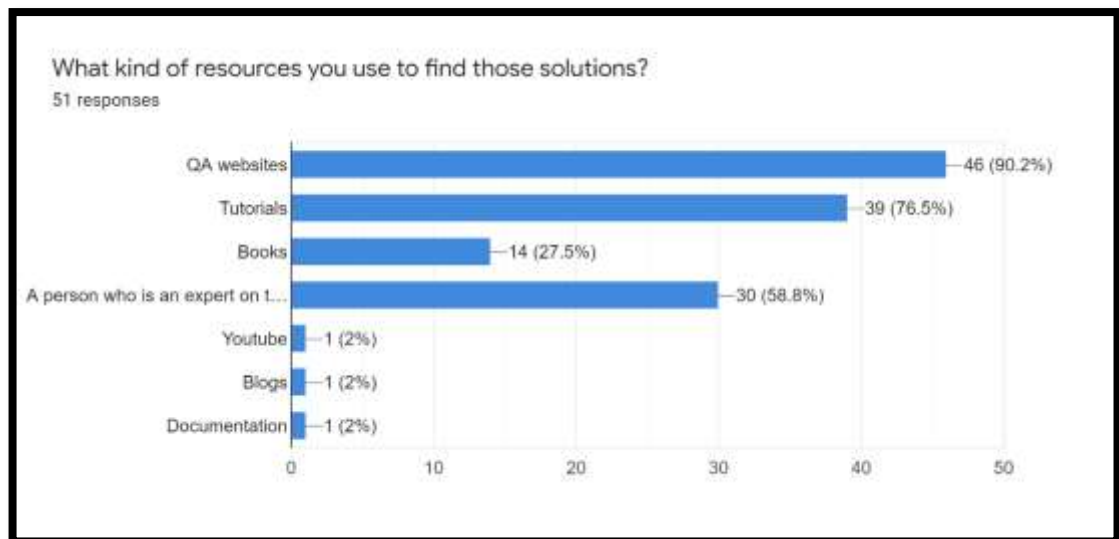


Figure 1.1 - Summary of responses for what kind of resources people used to find the solutions.

The other fundamental questions were then based on the amount of time undergraduates spent searching for answers, figure 1.2 shows that the majority of responses lasting more than 30 minutes. This implies that the average person must spend a significant amount of time searching for a solution because it is not easily found in one go. This could lead to dissatisfaction if they had to visit more websites than they planned to because they couldn't find the right answer to their question.

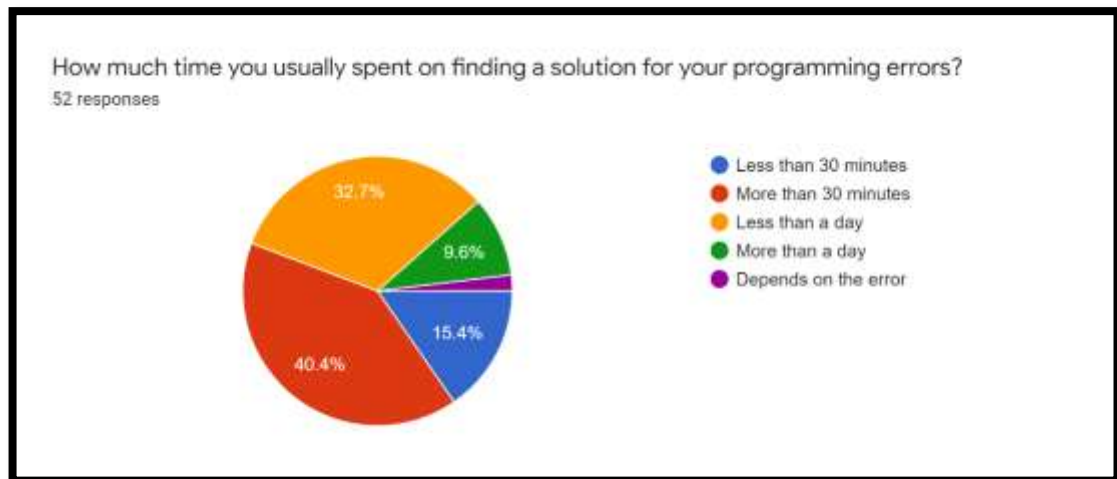


Figure 1.2 - Summary of responses for how much time spent on finding a solution for errors.

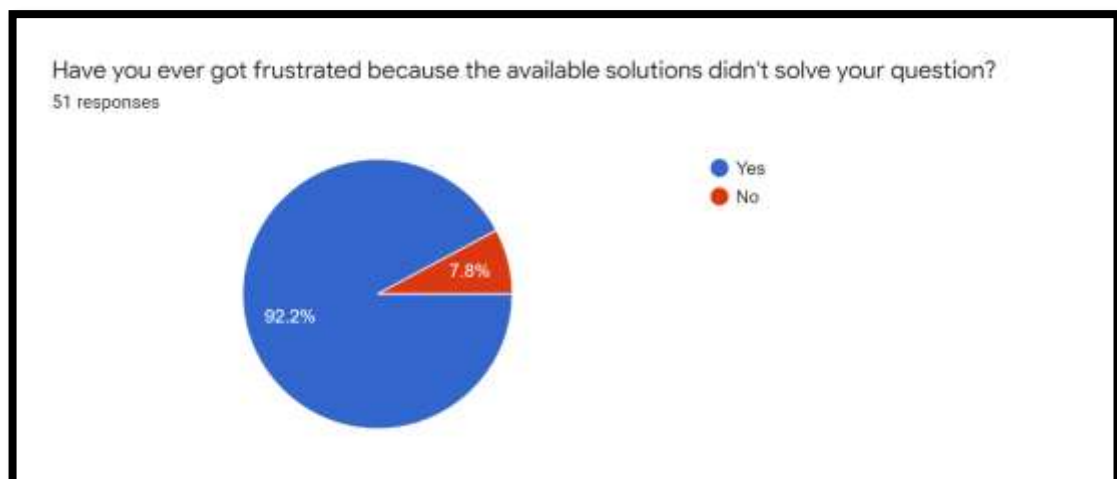


Figure 1.3 - Summary of responses for getting frustrated because a available solutions did not solve the questions.

The advanced question were mostly based on to find what the undergraduates desire when finding solution for their problems. As shown in Figure 1.3 users get much more frustrated as the answers are not reliable and waste their time. So as shown in Figure 1.4 people have to try all the links pop up on the search. Also, sometimes available solutions are not working. This could be because of a version mismatch; a library was not mentioned or any other missing part. However, figure 1.5 shows that how hard it is to get all the info regarding the particular problem in one try.

Another reason is availability of multiple solutions that work. As shown in Figure 1.6 users get confused and sometimes try all the solutions even though one worked but they are not sure.

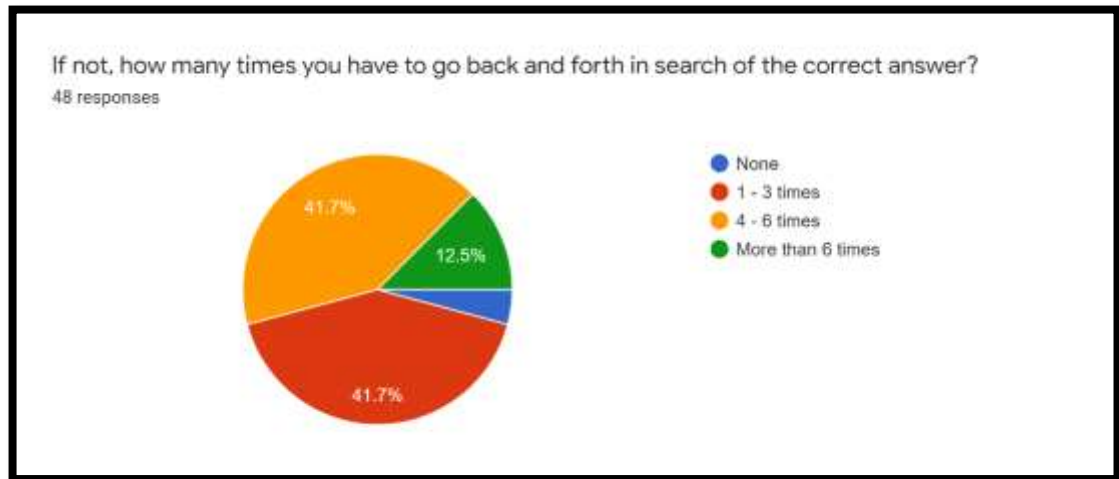


Figure 1.4 - Summary of responses for how many times users have to go back and forth in search of the correct answer.

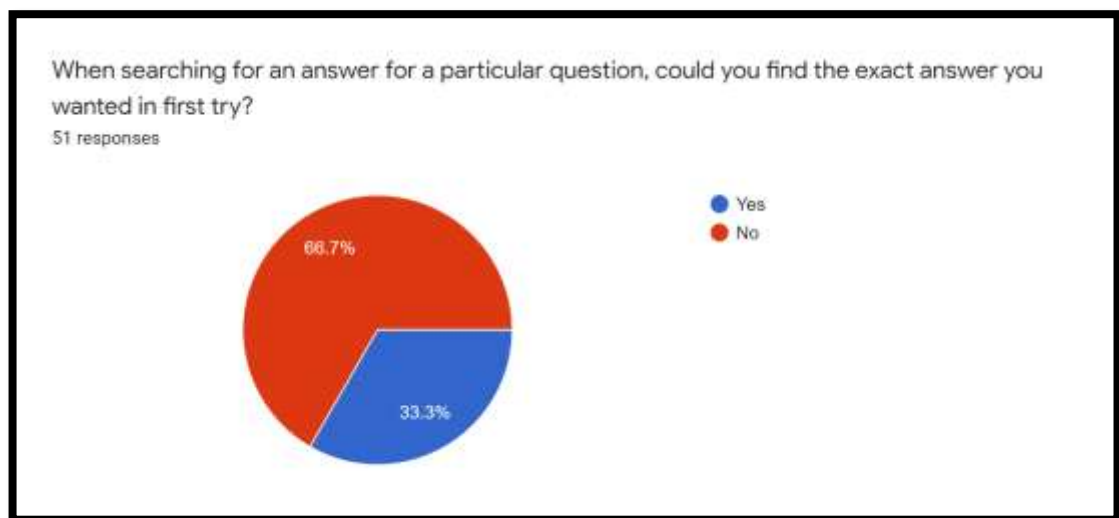


Figure 1.5 - Summary of responses for finding the solutions they wanted in first try.

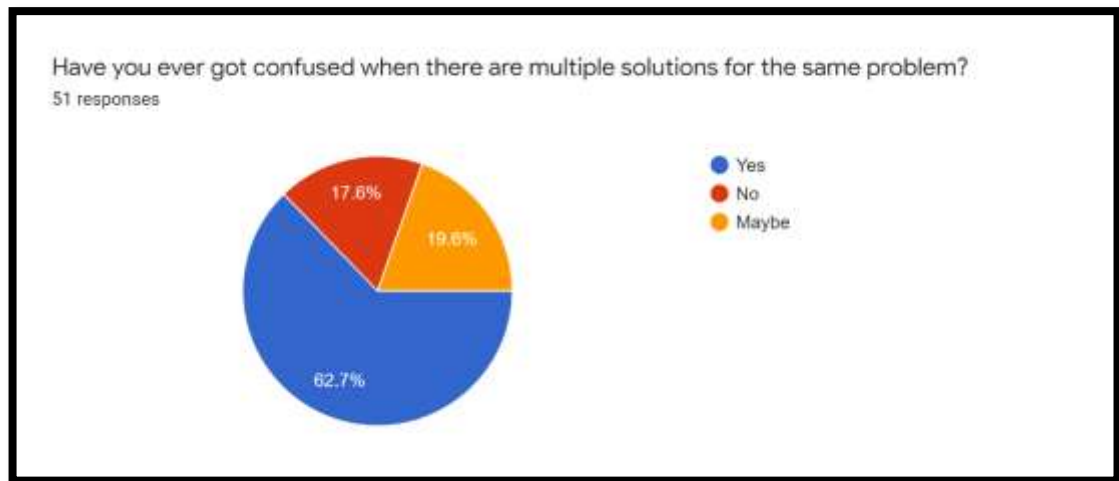


Figure 1.6 - Summary of responses for getting confused due to availability of multiple solutions.

The vast majority of students said that they require some kind of automated method to give direct replies to an online question or query. They give the following reasons: (1) it's sometimes difficult to describe the problem they're having, so some hints would be helpful; (2) there's too much noisy and repetitive content online; (3) answers in long articles are difficult to find; and (4) even the solution they find may just cover one area of concern. The answer generating tool is expected to generate a succinct and diverse summary of potential responses, which will assist in comprehending the problem and refining the queries/questions, according to the developers.

According to the results of our survey, there is a significant demand for new approaches for information retrieval and exploration. Instead of simply delivering answer ports with answers, we hope to develop an automated technique for generating answer summaries to developers' technical queries in this effort. Many technical questions posed by developers are non-factoid in nature[6], for example, what are the differences between a HashTable and a HashMap, for example?, In Java Swing, how can I create logs and have them shown in real time? For non-factoid technical inquiries, the response summary may consist of several sparse and distinct phrases.

'ProbExpert' is developed to address these concerns, which is more advanced and more equipped to handle challenges than the current Q&A systems available. As a public dev platform, anyone can answer the questions posted here, but the downside of this is that one question can get several duplicate answers, which inevitably leads to wasting the time of the person, and that is an area of focus with this research, 'reducing the time spent on finding the solution.' Therefore, to reduce this type of redundancy, an optimized answer is displayed. This process is focused on bringing all the alternative solutions into one optimized answer without harming the semantic information. Overall, this thesis covers how acquire optimized responses and generate automatically summarized answers from the publicly collected information.

1.2 Research Gap

When creating an optimized answer from a set of given answers has some positive and negative effects on different criteria such as.

- Summarization
- Answer quality
- Merge answers into a single answer

1.2.1 Automatic Text Summarization

Summarization is the process of compressing a piece of text into a shorter version, lowering the size of the original text while maintaining important informative components and content meaning. Because manual text summarizing is a time-consuming and typically hard process, automating the work is gaining popularity and thus serves as a significant motivator for academic study [7],[8].

Automatic text summarization is a difficult subject that is gaining traction these days. When an input is applied, the automated summarization produces an automatically summarized output. Although research on Automatic Text Summarization began in the 1950s at IBM Research Laboratories the area of Text Summarization has witnessed exponential growth in recent years as a result of the Internet[9]. Because there is so much information on the Internet, it is quite difficult to manually summarize enormous amounts of material. On the other hand, the Internet is a vast

library that contains far more information than is required. As a result, it is critical for looking for important papers among a large number of documents available. The goal of text summarizing is to condense the source material into a smaller version while retaining its information value and overall meaning.

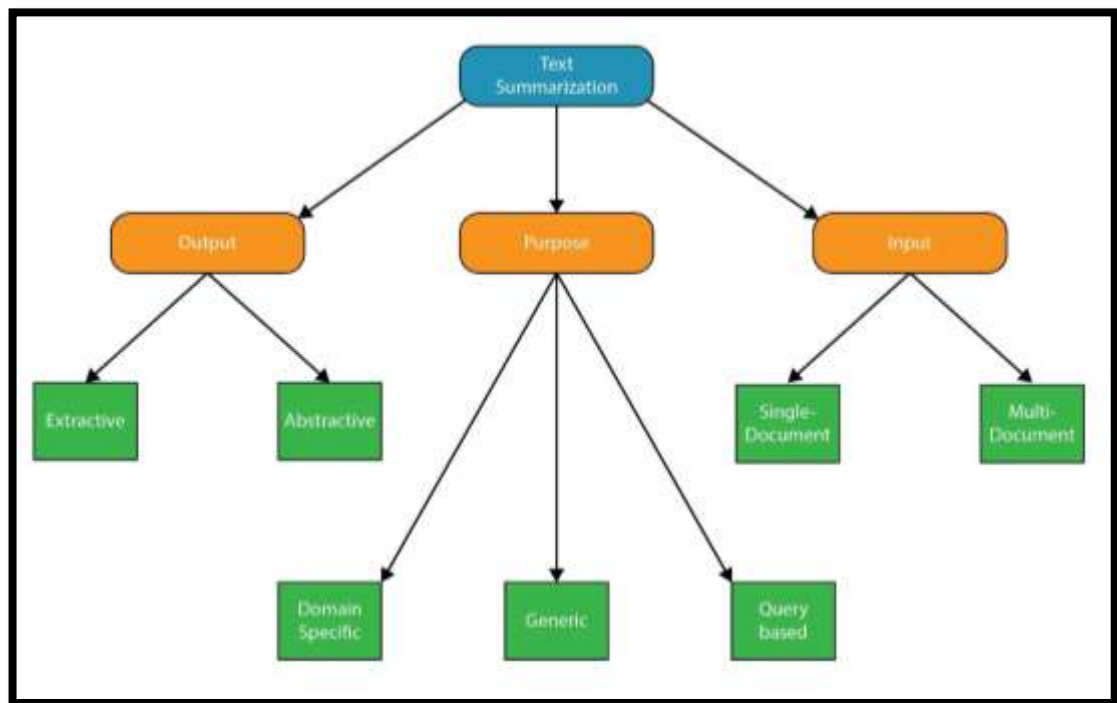


Figure 1.7 -Automated text summarization approaches

When considering summarization based on the output, there are 2 types as extractive and abstractive summarization. Abstractive Summarization is a way for creating new phrasing that represents the content of a text using natural language processing heavy equipment such as grammars and lexicons. Extractive summarization is a technique for identifying key text units (typically sentences) by looking at their lexical and statistical importance or matching phrasal patterns[10]. Extraction approaches are simple to adapt to larger sources, but the resulting summaries may be incoherent. Abstraction procedures produce sophisticated summaries and adapt well to high compression rates, whereas extraction approaches produce complex summaries and adapt well to high compression rates. Extractive summaries are made by extracting

important text segments from the text, such as sentences or sections, and evaluating the statistical data of solo or combined surface level attributes[11].

1.2.2 Extractive approaches for Text Summarization

To tackle challenges with artificial text summarization, two ways have been proposed. Extractive approaches, for example, try to choose a subset of sentences from source text documents. This procedure can be thought of as determining the most important sentences in original text materials.

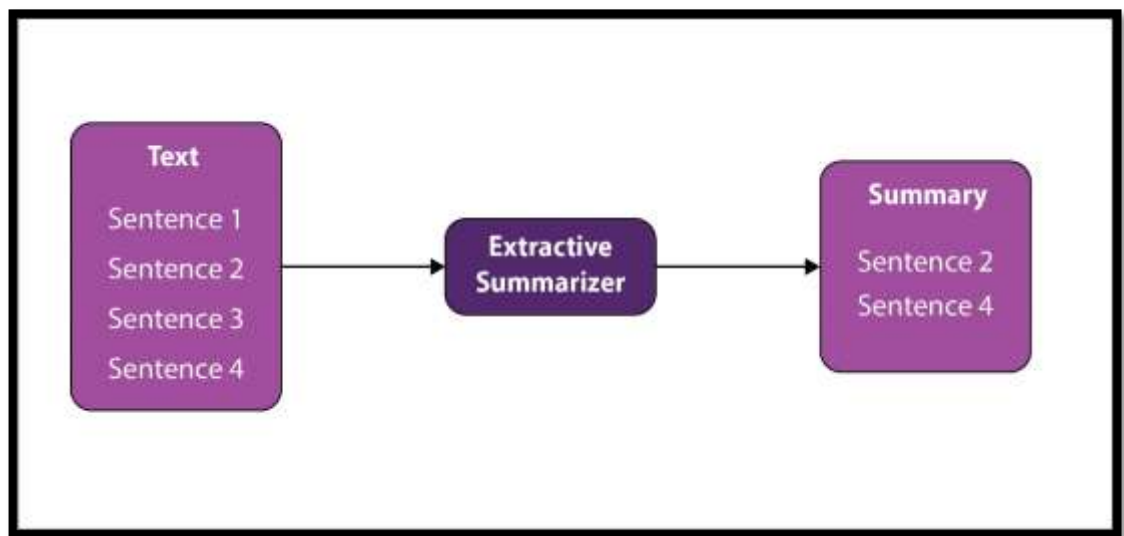


Figure 1.8 - Extractive summarization

The importance of a phrase is frequently measured by the centrality of the words that make it up. A common method of determining word centrality is to use the centroid of the document cluster in a vector space. The centroid of a cluster is a pseudo-document made up of words with tf idf scores greater than a preset threshold, where tf is the cluster's frequency and idf values are often computed across a much wider and related genre dataset. In centroid-based summarization, sentences with more words from the cluster's centroid are deemed central[12].

Later, Erkan and Radev[13] created LexRank, which computes sentence relevance using the concept of eigenvector centrality in a sentence graph representation. As the adjacency matrix of the graph representation of sentences, a connectivity matrix based on intra-sentence cosine similarity is used.

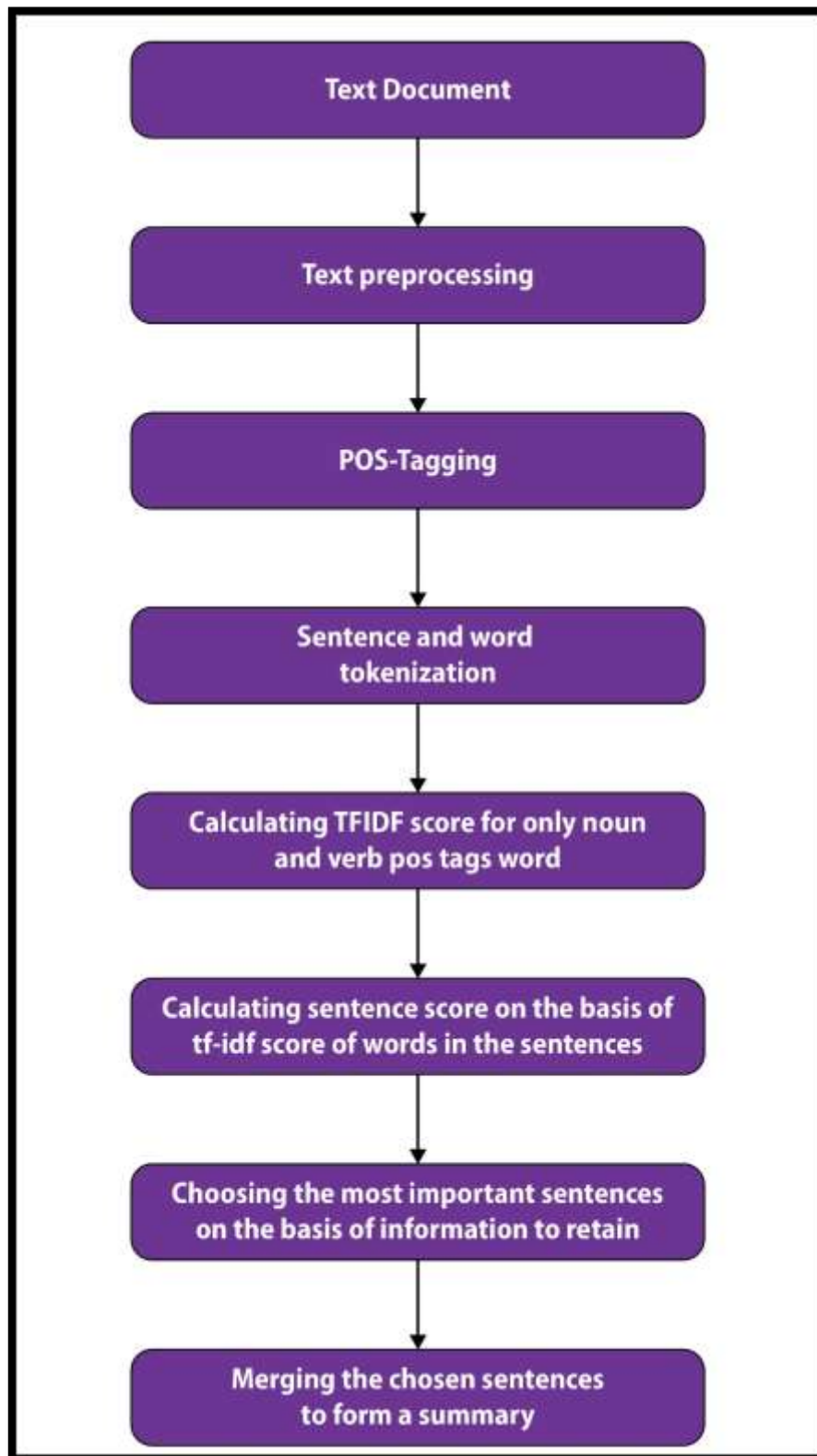


Figure 1.9 - Extractive summarization with TF-IDF

1.2.3 Abstractive approaches for Text Summarization

A substantial body of work on traditional text summarization has been undertaken, with the goal of identifying key lines or passages in the source document and reproducing them as a summary[14], [15]. Humans, on the other hand, tend to retell the original story in their own terms. As a result, human summaries are abstract in nature and rarely consist of reproducing original sentences from the document.

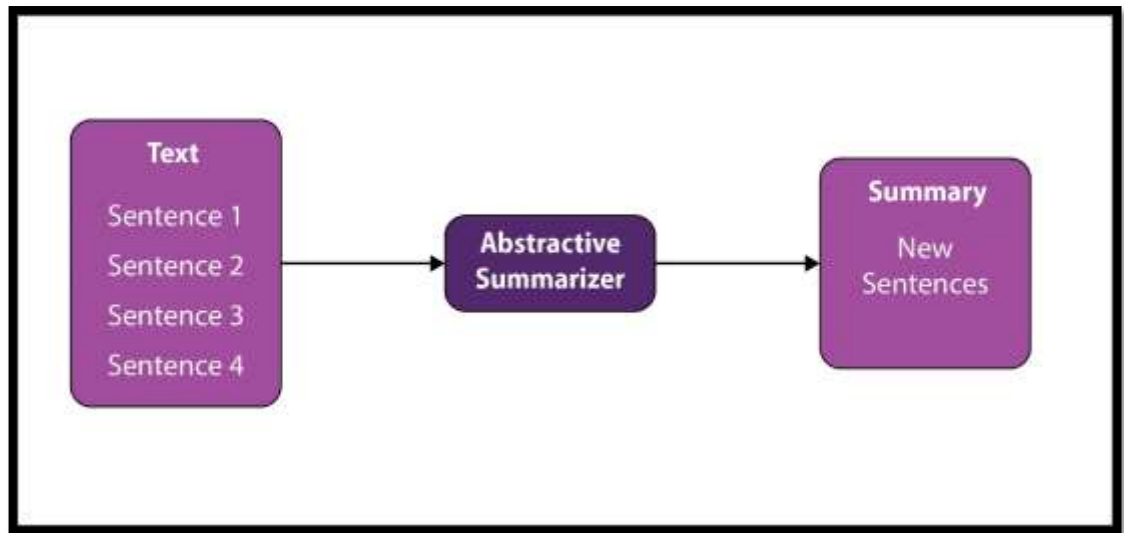


Figure 1.10 - Abstractive summarization

Extractive summarization is easier than abstractive text summarization because it does not necessitate complex natural language creation techniques. To make sense of the generated summaries, specialized linguistic patterns and subject knowledge are required, which takes a long time. As a result, there is still a significant grammatical and original quality difference between human-generated summary and abstractive summary.

Deep learning approaches and their uses in natural language processing tasks have recently emerged to aid in abstractive text summarizing tasks. To tackle this difficulty, a few recent works use an encoder-decoder system. Through an end-to-end system, such a framework seeks to map a pair of input text sequences to another output text sequence. Deep neural networks are used as encoder/decoder

components. Rush et al.[21] used a CNN to encode the source document and a context-sensitive attentional feed-forward neural network to create the summary.

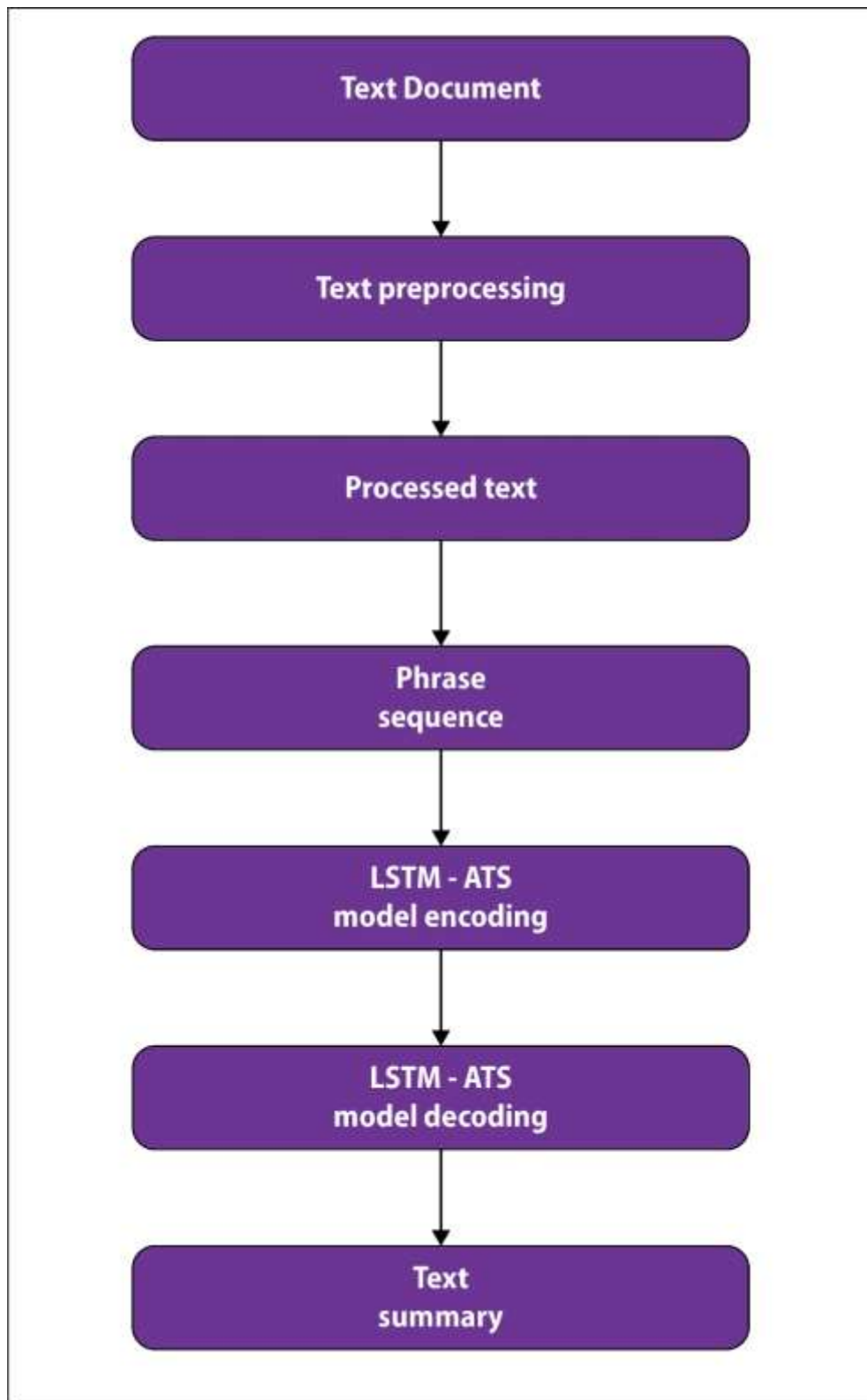


Figure 1.11 - Abstractive summarization with LSTM

Chopra et al.[22] employed a CNN for the encoder and a RNN for the decoder. Following the achievement of using the RNN Encoder-Decoder system to machine interpretation tasks, researchers began to employ the same architecture for abstractive text summarization. Lopyrev[23] to produce news headlines, a selective attention RNN Converter with LSTM units is used. For short text summary in Chinese, Hu et al.[24] use an RNN Encoder-Decoder. Nallapati et al.[25] revise the attentional RNN Encoder-Decoder framework by,

- 1) adding POS, NER, and TFIDF values to the encoder's input
- 2) In the decoder's output vector, unseen words are replaced with words from the input sequence.
- 3) collecting attentional alignment at the word and sentence levels between the upstream and downstream text sequences.

The authors use this paradigm on the DUC corpus and get significant results when compared to baselines.

The authors use this paradigm on the DUC corpus and get significant results when compared to baselines.

1.2.4 Question And Answer Quality

Because there are so many Q&A websites, delivering high-quality material is vital for a Q&A website to stand out. The relevance of high-quality content in community-driven question and answer websites has been recognized and investigated in several research. Importantly, Agichtein et al.[26] demonstrated a link between question quality and answer quality, implying that question quality influences Q&A service quality. In response to excellent queries, good responses are more likely to be given. In the same way, erroneous responses appeared in response to improper requests according to Agichtein et al.[26]. High quality questions, according to Li et al. [27], are predicted to garner more user attention, have more answer attempts, and become the best answer in a short amount of time.

As noted in the preceding data, there are two basic types of summarizations approaches accessible today. After extensive testing, we determined that abstractive summarization, rather than extractive summary, would be the most effective and human-friendly approach for this study. We also focus on how to achieve a quality answer, which lends credibility to both the answerer and the community as a whole. It also motivates users to ask more questions because they are confident, they will learn something new. In terms of the answer merging procedure, no research has been conducted to determine how to merge alternative solutions and obtain an optimal result.

The table 1.1 shows a comparison between our system and available systems and how effective our systems will be.

Table 1.1 - Feature comparison of Q-and-A platforms

Platform	Features		
	Answer Up-vote System	Provide optimized answers	Answer comparison
Quora	Yes	No	No
Stack overflow	Yes	No	No
ProbExpert	Yes	Yes	Yes

1.3 Research Problem

The main objective of this thesis is to provide an indicative summary for answers that have been upvoted for certain questions. In order to be utilized as a summary to describe an optimized answer, the answers must satisfy three requirements.

- **Relevancy:** The answer should reflect the primary issues mentioned in the questions
- **Answerability** The answers given by users must be relevant to the question
- **Diversity:** The information conveyed by the questions is meant to be non-redundant.

In the sections that follow, we'll look at three sub-questions in order to solve this problem while meeting the aforementioned goals.

1.3.1 Research Question 1: Where to find the questions and answers?

Now that we're going to use upvoted responses to create a summarized answer, we need to determine where the questions and answers are coming from. Is it possible to extract questions and answers from original text documents? Can they be drawn from a pre-existing question corpus/database? Can they be generated entirely automatically?

The most straightforward option is to utilize a database that contains questions and answers. One option is to add a new Q&A service database to the platform we're launching, while another is to take use of the rising popularity of Question and Answer (Q&A) services like Quora and Stack Overflow, whose Q&A databases may now be utilized as candidate question sets for summary development. The questions and answers in Q&A databases are given by real-world users, thus they are representative of real-world questions.

1.3.2 Research Question 2: How to evaluate the effectiveness of an optimized answer?

There is no ground-truth dataset for evaluation because using answers to summarize into an optimized answer is a novel task. We could request a user to read a series of questions and responses and then construct a summary based on the replies. We believe that a person can successfully identify the most essential portions of a question and construct an answer-based summary. If an automatic technique's set of

responses has a high overlap with the human-generated summary, the automatic method should be considered effective.

1.3.3 Research Question 3: How to save time on use of an optimized answer?

One of the key questions we addressed was how to reduce the amount of time spent searching for a solution on an online platform. Because time is increasingly valuable in today's world, finding a solution as soon as possible is essential. In this thesis, we focused on how this optimized solution may save time. Typically, when using a Q&A platform, a user must browse through numerous answers before finding a viable solution. But by using an optimized answer, the user must be able to get the solution in one search.

In summary, the purpose is to 1) decide whether to use an existing Q&A database or to establish a new Q&A database; 2) develop a criterion for evaluating generated optimized answers; and 3) Save time when searching for a solution.

1.4 Research Objectives

1.4.1 Main Objective

To eliminate the problems stated above, an e-learning platform powered by Machine Learning and Artificial Intelligence is proposed, enabling personalized learning path creation for users, and determining individual learning to be more resourceful and compelling. The ML-powered platform has the capability to find answers to users' subject related problems/questions across the internet archives. This facility is a great help for users as this process saves valuable time of the user by bringing answers and references to a single place with more accurate information. If not satisfied, users are given the option to ask questions in the platform's thread section to get answers from the respective experts. The system will be intelligent to generate an optimal answer from up-voted answers to form a complete answer. Additionally, with the help of previously answered questions, platform offers a quiz option to either refresh or

solidify users' knowledge on their subject of interest. Apart from the above features, a user can get support from an expert on their field through video conferencing as well. The platform has the ability to accurately measure users' proficiency by evaluating the users' contribution to other platforms using machine learning to rank them in the platform. This evaluation method is a great opportunity for the users as well because this process generates a valuable portfolio of the user which can be used to showcase their skills/talents to the outside world.

1.4.2 Specific Objective

In order to reach the main objective, specific objective mentioned below has to be fulfilled.

Generating an optimized answer using the top voted answers

- Check and remove similar answers
- Key word extraction
- Text preprocessing
- Summarize answers individually
- Merge answers together
- Check quality of the merged answer

In the ProbExpert platform users can openly post their questions in the thread section to get answers from different expert users. Once a question gets several answers, the 4 top voted answers will be selected to formulate an optimal answer for the viewers. Analyzed answers will be processed to check whether answers share similar content. If answers share similar content server will merge them to reduce redundancy. Then the selected answers will go through a preprocessing process. After that the summarization takes place by summarizing and merging all the remaining answers into a one answer. Then the server will save the answer marking it as a formulated answer. Then it will be shown as the optimal answer to the question.

2. METHODOLOGY

2.1 System Overview

To archive end result, ProbExpert platform contains two backend servers, one node JS and one python server, one cloud hosted MongoDB database, and to achieve the end goal, there are five Python models for data retrieval, text preprocessing, answer summarization, answer similarity checking and answer merging models. The back end will be connected with the front end through a REST API. Overview of System diagram is shown in figure 2.1

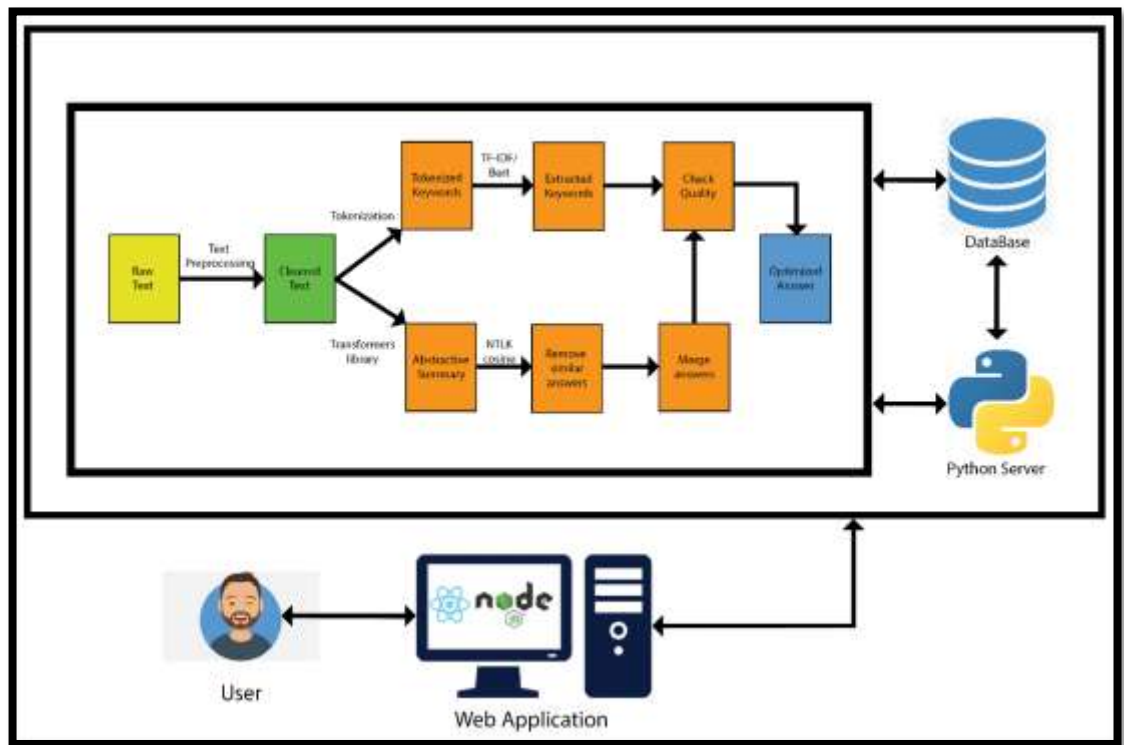


Figure 2.1 - System Diagram

2.1.1 Data Retrieval

The database constructed to contain all of ProbExpert's data will be the primary source of data for the functionality of this entire component. This database is powered by MongoDB, and we'll need to utilize queries to get the information we

need. The questions and answers will be extracted with their respective details using an extraction tool. If there are more than 5 responses, only the top four votes will be taken onto the next step. MongoDB is a NoSQL database that instead of the traditional table-based database structure stores data as JSON-like documents with customizable schemas. The document model is linked to objects in the application code, making it easier to deal with. It includes a rich query language that allows for dynamic inquiries on documents. MongoDB also offers its own aggregation pipeline and map-reduce functionality, removing the need for complex data pipelines. PyMongo is a Python package that includes utilities for working with MongoDB databases. PyMongo was used to query and get the essential data needed for this research.

2.1.2 Key word extraction

It's a technique for analyzing text. We can learn a lot about the subject in a short amount of time. It assists in the condensing of information as well as the identification of important keywords. It saves time because it eliminates the need to read the entire document. Two use-cases are identifying interesting topics from a news item and recognizing problems based on customer ratings. One of the methodologies used for Keyword Extraction is TF-IDF (Term Frequency – Inverse Document Frequency).

ML, AI and NLP are used in keyword extraction to break down human language so that it can be interpreted and evaluated by machines. It is used to extract keywords from a wide range of material, including conventional documents and business reports, social media comments, internet forums and reviews, news items, and more.

TF-IDF

TF-IDF is widely employed in machine learning algorithms for a variety of purposes, including stop-word elimination. These are terms like "a, the, an, it" that appear frequently yet provide little information. TF-IDF is made up of two parts: term frequency and inverse document frequency.

Term frequency can be calculated by counting the number of times a term appears in a document.

The IDF is computed by dividing the total number of documents by the number of documents in the collection that include the phrase. It's beneficial for decreasing the weight of terms that shows often in a set of text. This figure's log is used to soften the effect of IDF.

2.1.3 Check and remove similar answers

In this step the top voted answers given to that question will be compared with each other. If they have a similarity, we will eliminate them. To do this we will be using cosine similarity with natural language toolkit.

NLTK

Tokenization, parsing, categorization, stemming, tagging, and semantic reasoning are all included in the text processing packages. It also includes graphical demonstrations and sample data sets, as well as a recipe book and a book covering the NLTK's main language processing tasks. Steven Bird, Edward Loper, and Ewan Klein produced the Natural Language Toolbox, an open-source toolkit for the Python programming language designed for use in development and education.

It uses a hands-on approach to educate computational linguistics topics as well as Python programming fundamentals, making it suitable for linguists with limited programming knowledge, engineers and researchers who need to delve into computational linguistics, students, and instructors.

Cosine Similarity

The cosine similarity metric measures how similar documents are independent of their size. It calculates the cosine of the angle created by two vectors projected in three dimensions. In this situation, the two vectors I'm talking about are arrays containing the word counts of two documents[28]. How does cosine similarity differ from the number of common terms as a similarity metric?

The cosine similarity is favorable because even if two comparable texts are separated by the Euclidean distance due to size, they can still have a lesser angle between them. The greater the resemblance, the smaller the angle.

2.1.4 Text Preprocessing

Text processing is a technique used in NLP to clean text and prepare it for model creation. It is adaptable and comprises noise in a variety of forms, such as emotions, punctuation, and text written in numerical or unique character forms. We must address these major issues because machines will not understand if we merely ask for numbers. To begin with, certain Python modules ease text processing, and their clear, uncomplicated syntax provides a lot of versatility.

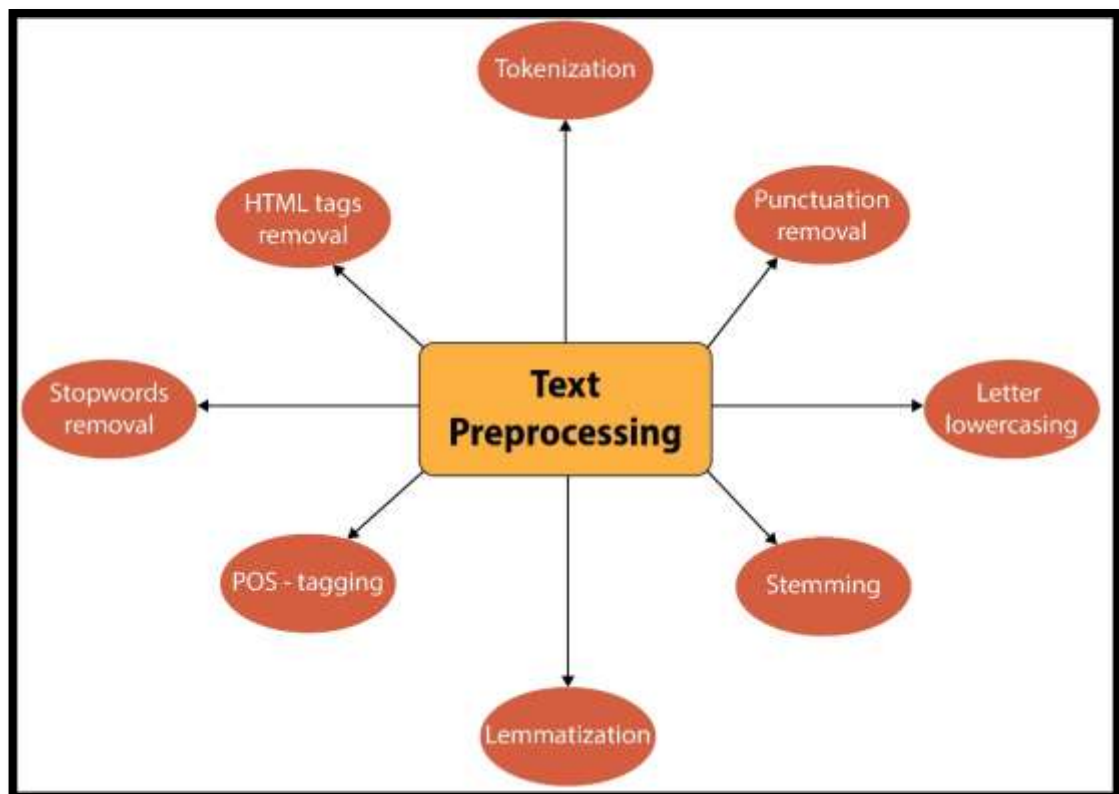


Figure 2.2- Text preprocessing

- Letter lowercasing

Lowercasing everything for the sake of simplicity is a typical practice. It aids in the maintenance of the consistent flow during NLP activities and text mining. The `lower()` method makes the entire process quite simple.

- Tokenization

Tokenization is a method of breaking down long strings of text into smaller bits called tokens. Tokenize larger sections of text into sentences, then sentences into words, and so on. Additional processing is frequently done after a piece of text has been suitably tokenized. Tokenization is also known as text segmentation or lexical analysis. Segmentation refers to the process of breaking down a large chunk of text into parts larger than words (e.g., paragraphs or sentences), whereas tokenization refers to the process of breaking down a large chunk of text into only words.

Key	Type	Size	Value
able	Float	1	0.16666666666666666
accepted	Float	1	0.003333333333333333
according	Float	1	0.2101010101010101
account	Float	1	0.5
accounts	Float	1	0.16666666666666666
across	Float	1	0.003333333333333333
action	Float	1	0.003333333333333333
active	Float	1	0.75
ad	Float	1	0.8
adamant	Float	1	0.003333333333333333
adamantine	Float	1	0.003333333333333333
adding	Float	1	0.003333333333333333
addition	Float	1	0.003333333333333333
adds	Float	1	0.003333333333333333
advising	Float	1	0.003333333333333333
aeschylus	Float	1	0.5
aether	Float	1	0.003333333333333333
afterlife	Float	1	0.003333333333333333
afterwards	Float	1	0.2
aid	Float	1	0.003333333333333333
aided	Float	1	0.003333333333333333
aignon	Float	1	0.003333333333333333
electo	Float	1	0.003333333333333333
allies	Float	1	0.16666666666666666
allowed	Float	1	0.10526315789473684
allusions	Float	1	0.003333333333333333
ally	Float	1	0.003333333333333333
along	Float	1	0.3333333333333333
also	Float	1	0.2101010101010101
although	Float	1	0.21052631578947367

Figure 2.3-Values given after Tokenization in a sample text

- Stemming

Stemming is a type of word normalization. It is a technique in which a group of words in a sentence are turned into a sequence in order to decrease the lookup time. Normalized words are those that have the same meaning but differ slightly

depending on the context or sentence. Stemming is thus a method of determining the underlying word from word variants.

- Lemmatization

Lemmatization is the algorithmic process of determining a word's lemma based on its meaning. Lemmatization is commonly used to refer to the morphological study of words with the goal of removing inflectional endings. It aids in returning a word's base or dictionary form, known as the lemma.

- Stop word removal

A stop words list is a collection of often occurring traits that can be found in any writing. Common features like or, and, but, and pronouns like he, she, it, and so on must be eliminated because they have no effect and add little or no value to the classification process (i.e., each feature should be removed when it matches any feature in the stop words list). If the feature is a special character or a number, it should be removed for the same reason. We can find stop words by sorting our keyword collection by frequency and selecting the most common ones based on their lack of semantic value.

Index	Type	Size	Value
0	str	1	i
1	str	2	me
2	str	2	my
3	str	6	myself
4	str	2	we
5	str	3	our
6	str	4	ours
7	str	9	ourselves
8	str	3	you
9	str	6	you're
10	str	6	you've
11	str	6	you'll
12	str	5	you'd
13	str	4	your
14	str	5	yours
15	str	8	yourself
16	str	10	yourselves
17	str	2	he
18	str	3	his
19	str	3	his
20	str	7	himself
21	str	3	she
22	str	5	she's
23	str	3	her
24	str	4	hers
25	str	7	herself
26	str	2	it
27	str	4	it's
28	str	3	its
29	str	5	itself

Figure 2.4- Stop words extracted from a sample text

- POS tagging

Part of speech tagging, often known as POS tagging or POST, is the process of categorizing words and labeling them according to their part of speech. To create sense in the sentence, a noun, pronoun, verb, adverb, article, adjective, preposition, conjunction, interjection, and a word must be fit into the right part of speech.

So, the part of speech is not only important in understanding the grammar of any language, but it is also an important aspect of text preprocessing in NLP, as we know that NLP is a task in which we produce a machine capable of communicating with a person or another machine. As a result, it becomes necessary for a machine to comprehend the POS.

2.1.5 Answer summarization

The preprocessed answers will be considered in the process of summarization. Natural language toolkit, BERT, hugging face's transformers library will be used to develop the summarization model. To summarize information, model needs related information. Predefined, programming related keywords will be supplied to fulfil that requirement.

BERT

BERT's core technical innovation is the application of Transformer's bidirectional training to language modeling. Transformer is a popular attention model. Researchers previously studied a left-to-right text sequence or a combination of left-to-right and right-to-left training. According to the findings, a bidirectionally trained language model can recognize language context and flow better than a single-direction language model. In the article, the researchers introduce a novel technique dubbed Masked LM (MLM), which allows bidirectional training in hitherto unattainable models.

BERT makes use of Transformer, which is an attention mechanism that learns contextual associations between words (or sub-words) in a text. Transformer consists of two separate mechanisms in its most basic form: an encoder that reads the text input and a decoder that generates a task prediction. Only the encoder technique is required because BERT's goal is to build a language model.

Several BERT (Bidirectional Encoder Representations from Transformers) based pre-trained models have been used in ProbExpert in order to maximize the expected results.

- a) *DistilRoBERTa base model*: The model consists of 6 layers, 768 dimensions, and 12 heads, with a total of 82 million parameters (compared to 125M parameters for RoBERTa-base). DistilRoBERTa is twice as fast as Roberta-base on average. OpenWebTextCorpus, a replica of OpenAI's WebText dataset, was used to train DistilRoBERTa.
- b) *Google's T5-base*: Transfer learning, in which a model is initially pre-trained on a data-rich job before being fine-tuned on an intermediate task, has emerged as a strong approach in natural language processing (NLP). The efficacy of transfer learning has produced a multitude of techniques, methodologies, and practices. The T5-base model is built utilizing the aforementioned concept, yielding cutting-edge results on a variety of benchmarks such as summarization, question answering, text categorization, and more. Monthly download average states close to 2 million, therefore indicating the effectiveness and the popularity.
- c) *bert-base-nli-mean-tokens*: The following model is a sentence-transformers model: It converts sentences and paragraphs into a dense vector space with 768 dimensions, which can be used for tasks like clustering and semantic search.

2.1.6 Answer merging

Following the summarizing process, the remaining text will be linked together to form an optimized answer. This will be accomplished by appending each summarized text to a single text document using basic Python code. The unified answer will then be sent to be reviewed for quality.

2.1.7 Check quality of optimized answer

The quality of the optimized answer will be checked using the previously extracted key words. If the comparison between extracted keywords and the summarized answer scores a higher value, it means that the semantic information in the original text has not been changed. If so, the optimized answer will be taken as the final answer and will be shown on the website.

2.2 Commercialization aspects of the product

Introducing ProbExpert into the market is a significant goal in this project. Below figure 2.2 illustrate the marketing plan and strategy we made for ProbExpert commercialization purpose.

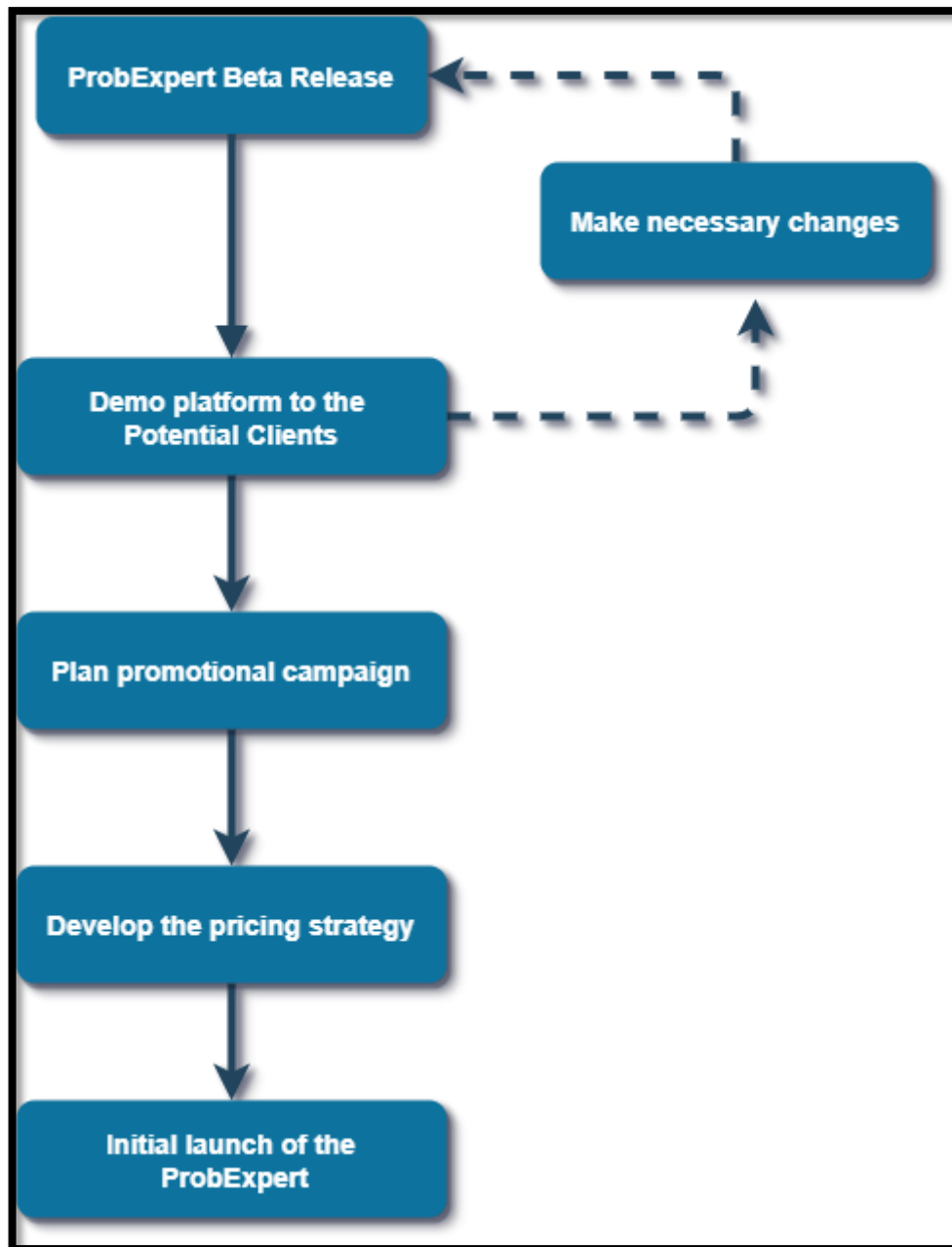


Figure 2.5 - ProbExpert's Marketing Strategy

Answer optimization option

As a commercialization aspect free ProbExpert users can only use the Answer optimization option twice per day. They would have to become premium members of the ProbExpert community if they wished to use the option more than twice. After becoming a premium member, he or she has unlimited access to the Answer optimization option.

Advertisements

Another commercialization option for the ProbExpert is to become an ideal advertising platform. Only a few areas relating to information technology, however, will be permitted to publish on the site. Allowed advertisements are divided into two categories, as shown here.

- a) **Product Advertisements:** Because the platform is inhabited with IT professionals, businesses can advertise their tech-related products or services on the site. Specifically, laptops, hosting and domains, and cloud storage. These personalized adverts have the potential to benefit both users and the platform.
- b) **Job Opportunities:** Organizations can list their available vacancies in the related section for jobs in order to attract great minds for the firm. Interested parties can then contact the company and participate in the business process. This service can be supplied as a pay-as-you-go service or for a fixed monthly or annual fee to display a limited number of advertisings based on the package selected.

2.3 Testing & implementation

How the system was tested under several testing techniques in order to minimize the issues are described in this section. It covers all the systems that are distributed inside the execution stage in detail how it totally was done.

2.3.1 Testing

The goal of testing is to find and correct flaws in the developed system. A tested programming system can be identified as a validated and approved system, and the testing process is critical to achieving the system. It is a method of verification and validation. The system is subjected to unit testing, system testing, and acceptance

testing and these will help to check whether the objectives are possible to achieve or not. A portion of the item objectives are,

- Real time processing
- Less time consuming
- Reliable and efficient
- Satisfy the necessities of the users' requirements.

a) Unit testing

Unit testing is the process of testing individual programs, subroutines, processes, and other components of a system. It is a method of handling the many components of testing. Unit testing should be completed for modules separately. Purpose of doing this is, it improves the unit performance without breaking and it decrease the expenses of repairing failures. This can be so useful because when a failure has been discovered it can be fixed properly since the testing is done separately. It eases the tasks of debugging. The system first developed in small programs which are integrated into the later phase.

b) Integration testing

In Integration Testing, the individual parts are altogether combined and also tested. The purpose of conducting the integration testing able to make verify the evaluation number of two or much more components produced the result to satisfy the functional requirement. This is done after the unit testing and before the validation testing. In here, testing of modules always started at the first level of the programming and continued to the lower levels.

Since the testing starts at in early stages of the implementation and failures can be caught earlier than its late in the development cycle. Easy to integrate due to the easiness to test in development environment. Comparing to the end-to-end tests, Integration testing run faster. Easy and more reliable to isolate the failures. For the Integration testing the developer and testers attitude required too.

c) System Testing

This compares the entire system or developed program with its original objectives. System testing is an attempt to identify the system fails to meet its objectives. It confirms not only the system design and development but also the users need as well. Integration testing passed outputs as taken as inputs in system testing. It identifies faults in both integrated parts and the entire system. The final result of this testing procedure is to examine the system's behavior. Basically, system testing is done by a testing team that is separate from the implementation team and helps to test the system's quality. It consists of both functional and non-functional testing.

d) Acceptance Testing

Acceptance testing is a process of comparing the system with its initial requirements and the current requirements of the end users. Usually this is performed by the customer or the end user. Developer will do the user testing before he handover the system to the user.

e) Regression Testing

Developers need to change or modify the functionality in the process, updates may occur in unexpected behaviors can have huge possibility. Regression testing normally performed to give that some change or part of addition has not change any of the existing function. In addition, its purpose is to find also bugs and errors that may have been occur in accidentally introduced into the existing solution and to verify that previously erased bugs continue to progress. Regression testing have many functional testing tools for continue their workload.

2.3.2 Implementation

The section covers all the system implementation which explained how the system works. Interfaces and the coding utilize for the better coordination and the development of the system. Number of testing's should perform in every stage to keep the system error free.

2.3.2.1 Interface implementation:

Web application

- The final outcome of this research will contain a web application to interconnect end users with the system. React JS will be used to build the web application. React JS is an open-source JavaScript library for creating single-page applications' user interfaces. In online and mobile apps, it's utilized to manage the view layer. React also makes it possible to create reusable user interface components. Next JS will be used along with React JS. Next JS is used to build server-side rendering and static web applications using React. Server-side rendering will be useful for search engine optimizations.

Web servers

- The final outcome includes web server to handle client sever communication, analyze data, gather information, process information, apply necessary algorithms, and save necessary information in the database. Server will be developed using python. HTTPS will be used as the communication protocol.

2.3.2.2 Machine Learning and Database implementation:

Database management

- To store data, system must need a database, NoSQL will be the best database type to store dynamic data. The database for this paper's results will be Mongo dB, which is a NoSQL database. MongoDB is a document-oriented database that uses JSON-like documents to store data. MongoDB's Sharding feature allows for horizontal scaling by dividing data over multiple machines and simplifying high-throughput operations with huge data sets.

Machine learning models

- Machine learning models will be used to find similar questions, extract keywords, and summarize information in this research paper. Relevant data sets will be gathered from the database created for this project to train and

perform the machine learning models. Specially Jupyter notebook and Python 3.8 was used.

2.3.2.3 Deployment requirements:

Digital ocean VPS

- The web application and the server must be on a publicly accessible server in order to general public to access the system. In order to host the web application and the server, Digital Ocean VPS (Droplets) will be used. Virtual Private Server (VPS) hosting is pooled server hosting that simulates server management environments. VPS hosting has grown in popularity as a cost-effective alternative to dedicated hosting while also providing superior dependability, security, and performance than shared hosting. Digital Ocean servers are easy to scale and have 100% network up-time and a 99.9% cloud uptime.

NPM – Node package manager

- The Node JavaScript framework has its own package manager, npm. It installs modules so that node can locate them and intelligently handles dependency conflicts. It can be designed to accommodate a wide range of scenarios. It is mainly used for releasing, finding, downloading, and creating node programs. npm will be used to handle modules in web application.

PIP – Python package manager

- PIP is a Python package management system that makes it easy to install and manage software packages. It provides a connection to the Python Package Index, an online repository of free and commercial Python packages. The Python Package Manager (PIP) will be used to manage Python packages on the web server.

3. RESULTS & DISCUSSION

In this section, results, and findings of the ProbExpert answer optimization system will be discussed.

3.1 Results

As initially ProbExpert database does not contain much data regarding questions and answers, a dummy dataset was used for the answer optimization process. Using the query shown in figure the top voted answers will be captured with the other essential detail. After that the retrieved data will be used for text preprocessing ,keyword extraction, summarization and so on. Figure shows the output of the query.

```
textcursor = list(collection.aggregate([
    {"$match": {"$and": [{"_id" : objInstance}, {"answers.2": { "$exists": "true" } }]}},
    {"$unwind": "$answers"},
    {"$sort": {"answers.score": -1}},
    {"$project": {"_id" : 0, "answers.text" : 1}}
]))
pprint(textcursor)
```

Figure 3.1- Sample answer extraction query

```
{'answers': {'text': 'When you use pull, Git tries to automatically merge. It is context sensitive, so Git will merge any pulled commits into the branch you are currently working on. pull automatically merges the commits without letting you review them first. If you don't carefully manage your branches, you may run into frequent conflicts.\n\nWhen you fetch, Git gathers any commits from the target branch that do not exist in your current branch and stores them in your local repository. However, it does not merge them with your current branch. This is particularly useful if you need to keep your repository up to date, but are working on something that might break if you update your files. To integrate the commits into your current branch, you must use merge afterwards.'}}
{'answers': {'text': 'The short and easy answer is that git pull is simply git fetch followed by git merge.\n\nIt is very important to note that git pull will automatically merge whether you like it or not. This could, of course, result in merge conflicts. Let's say your remote is origin and your branch is master. If you git diff origin/master before pulling, you should have some idea of potential merge conflicts and could prepare your local branch accordingly.'}}
{'answers': {'text': 'git fetch is similar to pull but doesn't merge. i.e. it fetches remote updates (refs and objects) but your local stays the same (i.e. origin/master gets updated but master stays the same). \n\ngit pull pulls down from a remote and instantly merges.'}}
{'answers': {'text': 'It cost me a little bit to understand what was the difference, but this is a simple explanation. master in your localhost is a branch.\n\nWhen you clone a repository you fetch the entire repository to your local host. This means that at that time you have an origin/master pointer to HEAD and master pointing to the same HEAD.\n\nWhen you start working and do commits you advance the master pointer to HEAD + your commits. But the origin/master pointer is still pointing to what it was when you cloned.\n\nSo the difference will be:\n\nIf you do a git fetch it will just fetch all the changes in the remote repository (GitHub) and move the origin/master pointer to HEAD. Meanwhile your local branch master will keep pointing to where it has.\n\nIf you do a git pull, it will do basically fetch (as explained previously) and merge any new changes to your master branch and move the pointer to HEAD.'}}
{'answers': {'text': 'One use case of git fetch is that the following will tell you any changes in the remote branch since your last pull... so you can check before doing an actual pull, which could change files in your current branch and working copy.'}}
{'answers': {'text': 'You can do a git fetch at any time to update your remote-tracking branches under refs/remotes/<remote>/. This operation never changes any of your own local branches under refs/heads, and is safe to do without changing your working copy. I have even heard of people running git fetch periodically in a cron job in the background (although I wouldn't recommend doing this).\n\nA git pull is what you would do to bring a local branch up-to-date with its remote version, while also updating your other remote-tracking branches.'}}
```

Figure 3.2-Output of the query

The one of main focus of this research was given to reduce time spent on searching for a solution in the Q&A platforms. However, in the ProbExpert platform the time taken for all the above-mentioned process and generating an optimized answer takes less than minute. In average the time developers spent on searching for multiple answers in the other Q&A platforms will be covered by this.

Table 3.1 - Sample time taken for answer generation

Process	Time(seconds)
Answer querying	10.8
Similarity checking	8.8
Text Preprocessing	5.6
Keyword extraction	9.2
Answer merging	4.3
Checking answer quality	12.5
Display answer to user	5.2
Total Time spent	56.2

3.2 Research Findings

Through our research, we discovered that a developer will meet a situation for which there is no answer on the internet or a Q&A platform at least once in his life. Even though it is not widely publicized, it is a problem that developers face on a daily basis. Furthermore, the time wasted by a new developer is substantially greater than that of a senior developer. This is because the book creator has a tendency to try the first option that appears as a solution on the internet, only to realize after a while that it was not the answer they were looking for. However, a significant amount of time has passed since the moment of realization. Nevertheless, by using optimized replies, developers will save a large amount of time because they will not have to try out every single solution or solution set.

3.3 Discussion

The system was tested in a situation where dummy data was used. These data used to confirm and verify the formation and the connection between the other components. And after then the system was tested in real environment. Dummy data was used here to confirm the actual working system and its components in real world scenario. This process is repeated over many times to configure for the feasibility of implementing the proposed system. The final outputs of the system confirmed the functions are accurate in real world environment as well.

3.4 Summary of the Student Contribution

Table 3.2- Student contribution

Personal	Functionality	Description
R.M.A.K. Ranasinghe	ProbExpert Platform	<ul style="list-style-type: none"> • System design • Database design • Backend structure design • API structure design

		<ul style="list-style-type: none"> • Overall UI developments • Common UI component developments
	Answer optimization	<ul style="list-style-type: none"> • Design user interface • Capture data • Create Training model • Generate an optimized answer

4. CONCLUSION

According to our early research, developers demand an automated answer generation tools to extract a clear and diverse summary of possible answers to their technical questions from the massive amount of information in Q&A exchanges.

To meet this need, we've developed a new Q&A platform that automates the development of answer summaries. Our user research shows that the automated answer summaries we generate are relevant, informative, and diversified.

This thesis focuses on a novel task: reducing a collection of replies to a single answer. A single answer is frequently more appealing to developers than reading a list of answers. To our knowledge, no previous study on text summarizing has looked into this problem in the literature, as most existing work on the topic focuses on extracting declarative phrases from original text sources or constructing declarative summaries.

REFERENCES

- [1] C. Speier, J. S. Valacich, and I. Vessey, “The Influence of Task Interruption on Individual Decision Making: An Information Overload Perspective,” *Decis. Sci.*, vol. 30, no. 2, pp. 337–360, Mar. 1999, doi: 10.1111/j.1540-5915.1999.tb01613.x.
- [2] D. Ravichandran and E. Hovy, “Learning surface text patterns for a Question Answering system,” in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics - ACL '02*, 2001, p. 41, doi: 10.3115/1073083.1073092.
- [3] C. S. Yadav and A. Sharan, “Hybrid Approach for Single Text Document Summarization using Statistical and Sentiment Features,” Jan. 2016, [Online]. Available: <http://arxiv.org/abs/1601.00643>.
- [4] B. Vasilescu, V. Filkov, and A. Serebrenik, “StackOverflow and GitHub: Associations between Software Development and Crowdsourced Knowledge,” in *2013 International Conference on Social Computing*, 2013, pp. 188–195, doi: 10.1109/SocialCom.2013.35.
- [5] K. Mao, Y. Yang, Q. Wang, Y. Jia, and M. Harman, “Developer recommendation for crowdsourced software development tasks,” in *Proceedings - 9th IEEE International Symposium on Service-Oriented System*

- Engineering, IEEE SOSE 2015*, Jun. 2015, vol. 30, pp. 347–356, doi: 10.1109/SOSE.2015.46.
- [6] H. Song, Z. Ren, S. Liang, P. Li, J. Ma, and M. de Rijke, “Summarizing Answers in Non-Factoid Community Question-Answering,” in *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, Feb. 2017, pp. 405–414, doi: 10.1145/3018661.3018704.
 - [7] E. Hovy, *Text Summarization*, vol. 1. Oxford University Press, 2012.
 - [8] I. Mani, D. House, G. Klein, L. Hirschman, T. Firmin, and B. Sundheim, “The TIPSTER SUMMAC Text Summarization Evaluation,” in *Proceedings of the ninth conference on European chapter of the Association for Computational Linguistics -*, 1999, p. 77, doi: 10.3115/977035.977047.
 - [9] H. P. Luhn, “The Automatic Creation of Literature Abstracts,” *IBM J. Res. Dev.*, vol. 2, no. 2, 2010, doi: 10.1147/rd.22.0159.
 - [10] U. Hahn and I. Mani, “The challenges of automatic summarization,” *Computer (Long. Beach. Calif.)*, vol. 33, no. 11, pp. 29–36, Nov. 2000, doi: 10.1109/2.881692.
 - [11] V. Gupta and G. S. Lehal, “A Survey of Text Summarization Extractive Techniques,” *J. Emerg. Technol. Web Intell.*, vol. 2, no. 3, Aug. 2010, doi: 10.4304/jetwi.2.3.258-268.
 - [12] D. R. Radev, H. Jing, M. Styś, and D. Tam, “Centroid-based summarization of multiple documents,” *Inf. Process. Manag.*, vol. 40, no. 6, pp. 919–938, Nov. 2004, doi: 10.1016/j.ipm.2003.10.006.
 - [13] G. Erkan and D. R. Radev, “LexRank: Graph-based Lexical Centrality as Salience in Text Summarization,” *J. Artif. Intell. Res.*, vol. 22, pp. 457–479, Dec. 2004, doi: 10.1613/jair.1523.
 - [14] H. Jing, “Sentence reduction for automatic text summarization,” in *Proceedings of the sixth conference on Applied natural language processing -*, 2000, pp. 310–315, doi: 10.3115/974147.974190.
 - [15] K. Knight and D. Marcu, “Summarization beyond sentence extraction: A probabilistic approach to sentence compression,” *Artif. Intell.*, vol. 139, no. 1, pp. 91–107, Jul. 2002, doi: 10.1016/S0004-3702(02)00222-9.
 - [16] P. Over, H. Dang, and D. Harman, “DUC in context,” *Inf. Process. Manag.*,

- vol. 43, no. 6, pp. 1506–1520, Nov. 2007, doi: 10.1016/j.ipm.2007.01.019.
- [17] D. Zajic, B. Dorr, and R. Schwartz, “BBN / UMD at DUC-2004 : Topiary,” *Proc. HLT-NAACL 2004 Doc. Underst. Work. Bost.*, pp. 112--119, 2004.
 - [18] M. Banko, V. O. Mittal, and M. J. Witbrock, “Headline generation based on statistical translation,” in *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics - ACL '00*, 2000, pp. 318–325, doi: 10.3115/1075218.1075259.
 - [19] T. Cohn and M. Lapata, “Sentence compression beyond word deletion,” in *Proceedings of the 22nd International Conference on Computational Linguistics - COLING '08*, 2008, vol. 1, pp. 137–144, doi: 10.3115/1599081.1599099.
 - [20] K. Woodsend, Y. Feng, and M. Lapata, “Title generation with quasi-synchronous grammar,” *EMNLP 2010 - Conf. Empir. Methods Nat. Lang. Process. Proc. Conf.*, no. October, pp. 513–523, 2010.
 - [21] A. M. Rush, S. Chopra, and J. Weston, “A Neural Attention Model for Abstractive Sentence Summarization,” Sep. 2015, [Online]. Available: <http://arxiv.org/abs/1509.00685>.
 - [22] S. Chopra, M. Auli, and A. M. Rush, “Abstractive Sentence Summarization with Attentive Recurrent Neural Networks,” in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016, pp. 93–98, doi: 10.18653/v1/N16-1012.
 - [23] K. Lopyrev, “Generating News Headlines with Recurrent Neural Networks,” Dec. 2015, [Online]. Available: <http://arxiv.org/abs/1512.01712>.
 - [24] B. Hu, Q. Chen, and F. Zhu, “LCSTS: A Large Scale Chinese Short Text Summarization Dataset,” Jun. 2015, [Online]. Available: <http://arxiv.org/abs/1506.05865>.
 - [25] R. Nallapati, B. Zhou, C. N. dos Santos, C. Gulcehre, and B. Xiang, “Abstractive Text Summarization Using Sequence-to-Sequence RNNs and Beyond,” Feb. 2016, [Online]. Available: <http://arxiv.org/abs/1602.06023>.
 - [26] E. Agichtein, C. Castillo, D. Donato, A. Gionis, and G. Mishne, “Finding high-quality content in social media,” in *Proceedings of the international*

conference on Web search and web data mining - WSDM '08, 2008, p. 183, doi: 10.1145/1341531.1341557.

- [27] B. Li, T. Jin, M. R. Lyu, I. King, and B. Mak, “Analyzing and predicting question quality in community question answering services,” in *Proceedings of the 21st international conference companion on World Wide Web - WWW '12 Companion*, 2012, p. 775, doi: 10.1145/2187980.2188200.
- [28] F. Rahutomo, T. Kitasuka, and M. Aritsugi, “Semantic Cosine Similarity,” *Semant. Sch.*, vol. 2, no. 4, pp. 4–5, 2012.