

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

2021-155

Marapana S.K.C.W.K.M.R.T.S.B.

IT18078992

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

November 2021

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

2021-155

Marapana S.K.C.W.K.M.R.T.S.B.

IT18078992

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

November 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

It is no secret that modern-day civilization is driven by knowledge. Therefore, the knowledge workers need to be informed about their respective vocations and the learnings. When we consider the programming area, this has a more significant impact as the technology is continually getting updated, and new technologies are appearing in short durations. Regarding the issue to be updated and informed about their earlier studies, the experts' proved and tested solution is conducting adaptive quizzes in repetitious time-space. By conducting quizzes this manner, the student can reinforce their previous knowledge and acquire exposure to new trends in the topic. It is more helpful for learners to focus on structured type questions as, unlike the MCQ-typed questions, Structured-type questions can dive into theories and implementations much deeper, helping to understand concepts better. [17] This strategy helps to strengthen the theoretical knowledge of the learners. As modern-day employers explicitly look into the theoretical knowledge in interviews, evaluating the knowledge through structured-type questions will aid undergraduates and fresh graduates. Moreover, having a proper way to analyze their answers and offer an appropriate and unbiased score for the effort is crucial for motivation and self-evaluation. Through this research, we are suggesting appropriate strategies to achieve the aforesaid scenarios.

Keywords- *theoretical quizzes, adaptive quizzes, structured type questions, knowledge checking.*

Acknowledgment

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, I would like to thank the Sri Lanka Institute of Information Technology (SLIIT) for according this opportunity to release innovative ideas through this project as a compulsory requirement of the course. 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 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 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 the least, I would like to thank all others whose names are not listed here, but have given their utmost encouragement and support in every possible manner.

TABLE OF CONTENTS

1	INTRODUCTION.....	1
1.1	Background Literature.....	1
1.2	Research Gap.....	2
1.3	Research Problem.....	4
1.3.1	Research Question 1: Reliability of the quiz answers	4
1.3.2	Research Question 2: Comparing of the answers	5
1.3.3	Research Question 3: Scoring method.....	5
1.3.4	Research Question 4: Adaptive questions	6
1.3.5	Research Question 5: Availability of a platform to evaluate the theoretical structured type questions.....	6
1.4	Research Objectives.....	8
1.4.1	Main Objective	8
1.4.2	Specific Objective.....	8
2	METHODOLOGY.....	10
2.1	System Overview	10
2.1.1	Question difficulty level identification	10
2.1.2	Keyword Extraction in answers.....	11
2.1.3	Summarization of the answers.	15
2.1.4	Comparing answers.....	17
2.1.5	Scoring method	20
2.1.6	Web Scraping for non-existing questions and answers.....	22
2.1.7	Pre-processing of the data.....	22
2.1.8	Models	25
2.2	Commercial Aspect of the Product.....	27
3	TESTING & IMPLEMENTATION	30
3.1	Testing.....	30
3.2	Implementation.....	32
4	RESULTS & DISCUSSION.....	35

4.1	Results.....	35
4.2	Research Findings.....	35
4.3	Discussion.....	36
4.4	Summary of the Student Contribution.....	37
5	CONCLUSION.....	38
6	REFERENCES	39

LIST OF Tables

Table 1-1 - Competitors comparison.....	3
Table 2-1: Stages of keyword generation.....	15
Table 2-2: Answer summarization.....	17
Table 2-3: Comparing answers.....	20
Table 2-4: Scoring model breakdown.....	21
Table 4-1: Student contribution.....	37

LIST OF FIGURES

Figure 1-1 - Usage of quiz platforms.....	2
Figure 1-2 - Reliability of receiving answers	5
Figure 1-3. - Quizzes and knowledge matching level	6
Figure 1-4 -Usage of platforms to check answers.....	7
Figure 0-1 - System architecture overview.....	Error! Bookmark not defined.
Figure 0-2: Cosine similarity formula.....	18
Figure 0-3: Tokenization process	23
Figure 0-4: Highlevel functionality of a BERT Transformer.....	27
Figure 0-5: ProbExpert's Marketing Strategy.....	28
Figure 3-1: FrontEnd deployment.....	34

LIST OF ABBREVIATIONS

Q&A	Questions and Answers
KE	Keyword Extraction
BERT	Bidirectional Encoder Representations from Transformers
REST	Representational state transfer
API	Application Programming Interface
ML	Machine Learning
NLP	Natural Language Processing
DL	Deep Learning
AI	Artificial Intelligence

1 INTRODUCTION

1.1 Background Literature

With modern-day knowledge shifting towards the internet, all learners can access vast information with several clicks. Even though learners have such privilege, it may be impossible to retain all the information in the brain, and it is not practical to rely on the internet as students or professionals. The best way to retain information is by taking quizzes in spaced repetition. Considering Roediger, Putnam, and Sumeracki (2011) research, the quizzes method is most effective. According to the paper, reading information and then consolidating and testing the quiz form's knowledge effectively helps retain the information[2]. Also, regarding a study based on students who span five years, the conclusion was that online quizzes had a proven positive influence on students' academics[1].

As mentioned above, while quizzes are a great tool to retain information in the brain, they directly impact learners' motivation and engagement[4] in their learning as per research conducted in 2018 using online quizzes. Considering the programming domain, which has a strong possibility of the variable of information, it is hard for individuals to memorize their critical points as they tend to be forgotten as new learnings come. These quizzes can be used as knowledge check questions in e-learning as these self-test questions can be used to assess learners' understanding of a subject and help eliminate misconceptions[3]. According to the survey conducted, figure 1.1 shows the use of online quiz platforms by the responders. As per the survey, most of the responders use quiz platforms to check their knowledge often.

How often do you use quiz platforms to check your knowledge?

71 responses

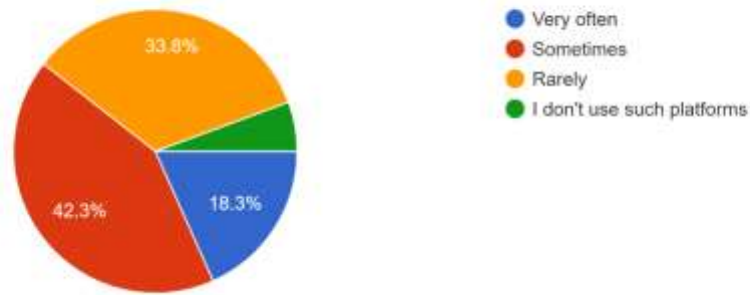


Figure 1-1 - Usage of quiz platforms

According to the research [1], the impact of using online quizzes is much higher on learning, and we can assume that the learners are also aware of this fact, thus the result of the above survey. Practicing the knowledge collected so far can be quickly done with quizzes, which helps tremendously with interviews, assignments, and examinations.

Using only a quiz-providing platform or similar implementation will not be helpful if the answers are not standard. As learners' answers will depend on each individual's writing capability, an unbiased method of evaluating the answers is needed to give everyone an equal opportunity. As per the research conducted on evaluating essay -typed answers, the evaluation's variance was significant [6]. This variance in marking is also actual in structured-type questions. Therefore as a solution for an unbiased evaluation method for user answers, Natural Language Processing (NLP) can be used [19] - [20] with similarity measurement techniques to achieve a better scoring method.

1.2 Research Gap

There are not many platforms right now for quizzes focused on programming on the internet. Platforms like StackOverFlow, Quora have a vast database of questions asked by the users and received proper, verified answers from experts. According to the research paper [7], only 10.9% of questions fall into unanswered categories. Experts of the fields answer the rest of the questions asked in the platform (89.9%), and having this massive data can be utilized for knowledge checking of the programmers, which area is ignored by the competitive platforms.

Another popular way of knowledge checking of programmers is measuring the ability to code. This method can be seen in many platforms such as HackerRank, LeetCode, or AlgoExpert. However, the problem is with the coding-focused approach; many critical theoretical parts of programming can not be discussed in-depth as most of these platforms focus on algorithmic and data structures areas, which are the base of competitive programming [18]. Furthermore, with this approach, many learners cannot get the full potential of a quiz platform. Hence the element of knowledge checking using structured-type questions is missing in the competitive platforms. HackerRank measures the given answers with the help of test cases, and if these test cases fail, there is no way of knowing how far the code was successful, and no marks will be given. That brings more pressure to the learner to do perfect coding to get the answer and move on. Therefore, some learners can be overwhelmed by this scoring method since HackerRank does not provide a proper scoring method for their effort.

Platform	Features			
	<i>Provide structured quizzes for knowledge checking</i>	<i>Provide good quality answers for users</i>	<i>Answer comparison</i>	<i>Scoring method</i>
StackOverflow	No	No	No	No
Quora	no	No	No	No
HackerRank	Coding questions only	Yes (Coding only)	No	Yes (With test cases, coding only)
ProbExpert	Yes	Yes	Yes	Yes

Table 1-1 - Competitors comparison

1.3 Research Problem

Many platforms only focus on polishing coding skills, and many of these learning aid platforms do not offer the facility to check knowledge on theoretical questions. Hence, through this research, we are trying to address how to check theoretical knowledge using structured type questions more accurately by offering adaptive quiz questions based on the users' knowledge level and introducing a better-unbiased grading method using similarity level. Below listed are the justifications of the sub-set questions of the leading research problem that has been identified.

1.3.1 Research Question 1: Reliability of the quiz answers

Another problem with knowledge checking using the quizzes is that there is no proper way to check the answered questions. As mentioned before, coding questions can be evaluated using test cases, and MCQ-type questions can be evaluated as there is/are fixed answer/s for the particular question, which might not be the same case regarding the structured-type questions because although the core of the answers stays the same expressing the answer can be unique hence a proper way to grade the answers is absent. According to the conducted survey, a majority (62%) responded that they had not used a platform to grade their written/typed answers to structured-type questions and were also not satisfied with the answers they had received. 71.8% of the responses were on a scale of 3 or below, where the max scale is considered 5, which is a mammoth proportion considering the targeted user group. As the research was done in 2012 [9], the researchers have identified why the internet's reliability is low. As they have identified three reasons and regarding the structured type questions, only two are applicable. One is the lack of explanations, and the other one is the shortcoming of the solution. With the inability to meet the above-stated requirements, the researchers recognized the answer reliability as low, and the impact is shown here through the survey as many are having second thoughts about the answer reliability, raising the first research question that needs to be answered.

In structured type questions, how reliable are the answers you receive ?

71 responses

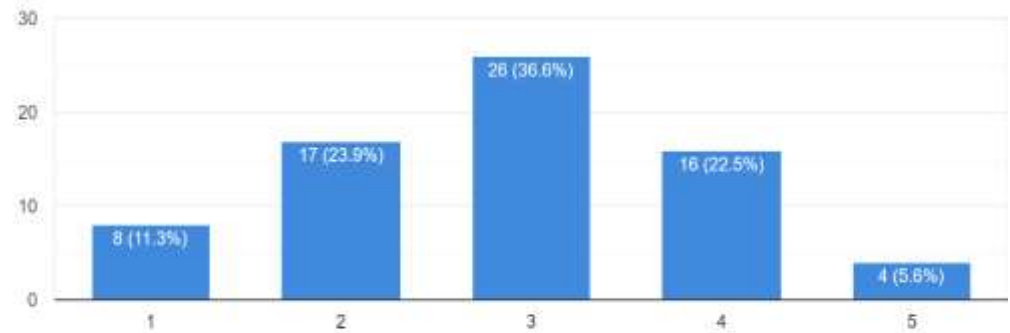


Figure 1-2 - Reliability of receiving answers

1.3.2 Research Question 2: Comparing of the answers

The second question raises the mechanism of comparing the user-provided answer and the accepted original answer of a particular quiz question. How can the introduced platform determine the similarity of both the answers? Is it considering the concrete or abstractive semantic similarity to determine the final output? What are the refined ways to accurately measure the similarity? These are the questions that arose while researching for a mechanism to compare the answers.

1.3.3 Research Question 3: Scoring method

A quiz platform's primary focus must be weighted on the quiz taker's final marking method output. Since almost all the available platforms such as HackerRank, LeetCode, or CodeSignal primarily focus on the test-case-based scoring model, therefore able to present a near-perfect score, the unavailability of a proper scoring method on theoretical questions is a significant concern trying to address via this research. Any newly introducing scorer evaluation model must be unbiased, not dependent on the users' answering style. Also, it

should be a model to satisfy every learner's input despite the user knowledge level.

1.3.4 Research Question 4: Adaptive questions

Most of the time, students/learners can be overwhelmed with the number of quizzes they encounter. Without adaptive quizzes based on each individual's knowledge level, these quizzes will either be too simple or too complicated, causing either dullness or discouragement [5]. Our survey data shows that more responders think their knowledge level is not tallying with the questions presented in online quizzes. This issue leads learners' to not select quizzes as a knowledge checking method since they do not provide enough value for their time. At the end of this research, the question that is trying to address here is how to present quizzes that are up to their knowledge level? Without having questions that are too easy or much difficult to answer, a smoother knowledge transfer.

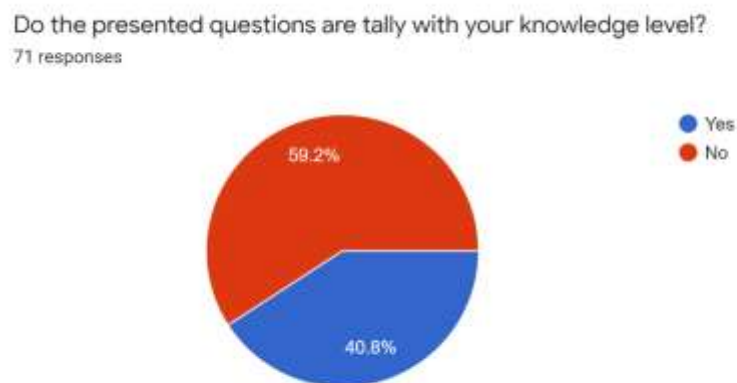


Figure 1-3. - Quizzes and knowledge matching level

1.3.5 Research Question 5: Availability of a platform to evaluate the theoretical structured type questions.

The last but crucial question encountered during the research is, 'Is there a platform available to evaluate structured type theoretical questions?' We have

encountered none regarding the research questions by comparing similar major and popular products in the Q&A platform business. In the previous sections, the spotlight was on platforms like StackOverflow, Quora, HackerRank, LeetCode, and CodeSignal. These platforms, especially StackOverflow and Quora, having access to possibly the largest programming questions database [31], have not considered the possibility of converting this valuable knowledge provided by millions of developers/experts to achieve the mentioned task, theoretical knowledge checking.

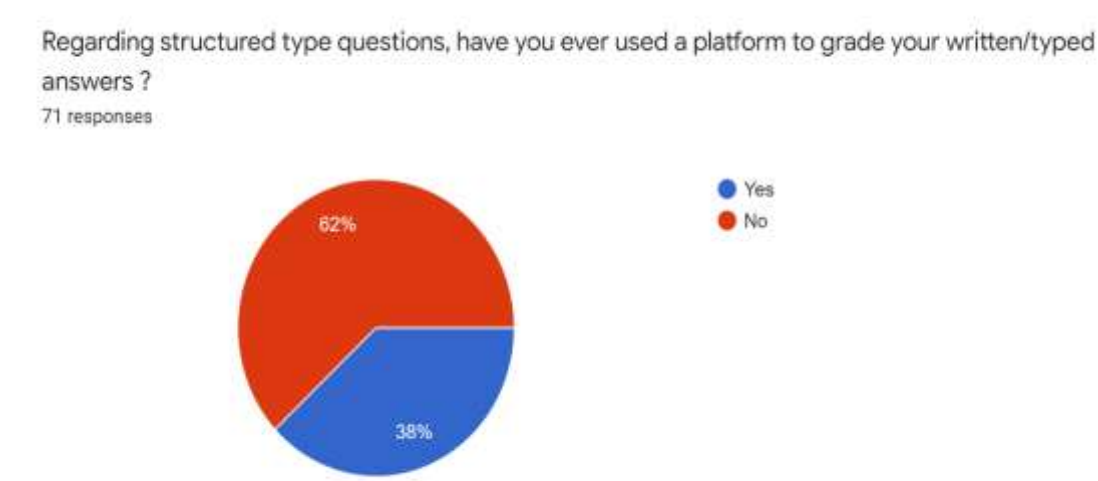


Figure 1-4 -Usage of platforms to check answers

1.4 Research Objectives

1.4.1 Main Objective

An e-learning platform powered by Machine Learning and Artificial Intelligence is proposed to eliminate the problems stated above, 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 the user valuable time 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 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, the 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 in their field through video conferencing. The platform can accurately measure users' proficiency by evaluating their contribution to other platforms using machine learning to rank them in the platform. This evaluation method is an excellent opportunity for the users because it generates a valuable user portfolio, showcasing their skills/talents to the outside world.

1.4.2 Specific Objective

Structured type quiz generation for knowledge checking by utilizing the answered platform question and introduce an unbiased scoring method for evaluating answers:

The user must select a specific topic/tag to sort out the related questions already posted on the topic to generate a quiz. Then identification of each of the questions' difficulty level, according to the users' expertise level, will be executed. (Refer section 2.1.1) If there are no questions about a user's topic, questions and answers will be formulated using external resources. (Section

2.1.6). Formulated questions will be structured-type questions. The system will use both generated optimal answers and the voted top four answers to increase the accuracy as the given answers will be different and mostly 'user unique.' Then the system will determine the similarity between the above answers and the user's answer along with keyword extraction and summarization techniques to assign a similarity score based on that metric, and final marks will be displayed. The answers getting from external resources will also go through the similarity checking process.

Therefore, the identified objectives are as follows:

- Formulating structured type questions for knowledge checking, using existing answered questions of the platform based on the user level.
- Formulate questions from the platform's existing user questions
- Formulate answers from optimal answer or top-voted answer and transform user's answer for similarity checking
- Similarity checking using cosine similarity and score assignment

2 METHODOLOGY

2.1 System Overview

The proposed system will consist of four machine learning models for the question-and-answer scraping, keyword extraction, answer summarization, and answer similarity checking models. Web crawlers will be used to scrape questions and answers that are not on our platform. The back end will be connected with the front end through a REST API.

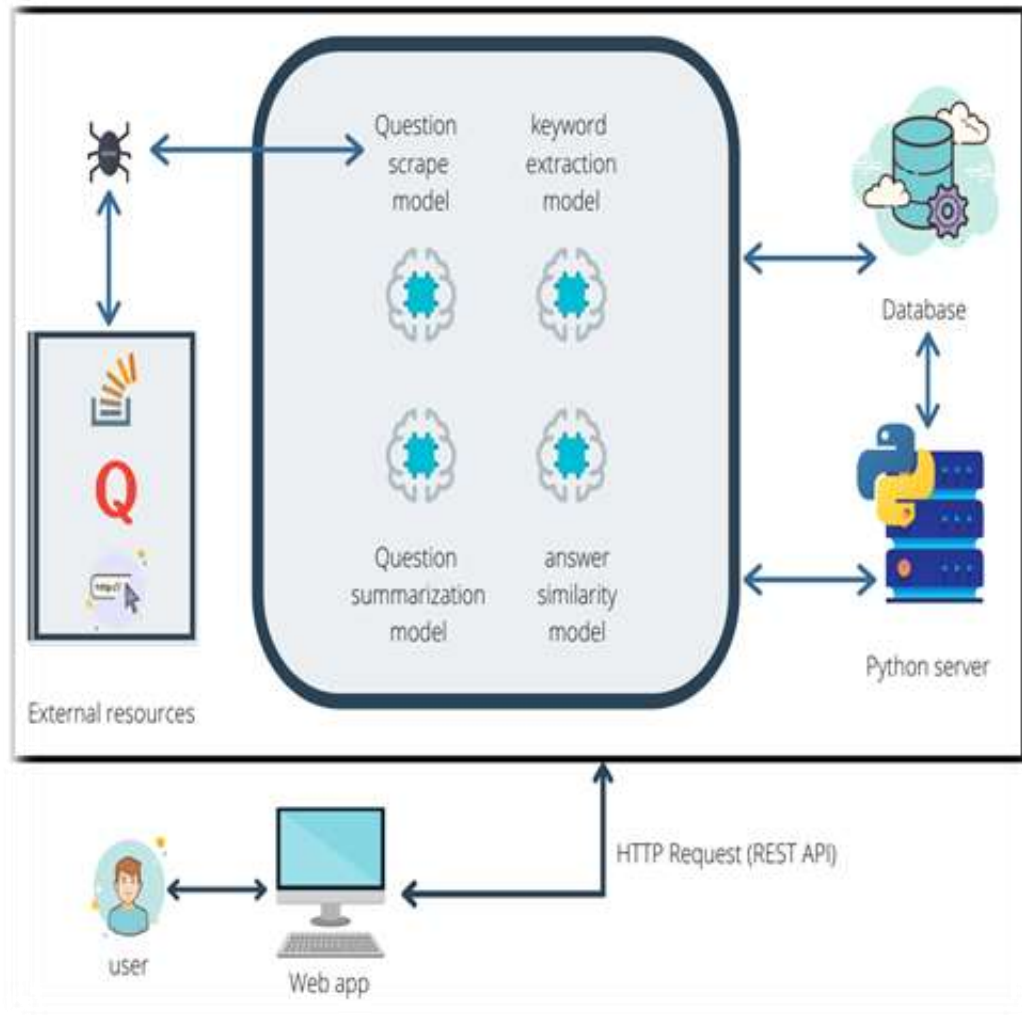


Figure 2-1: System architecture overview

2.1.1 Question difficulty level identification

Identifying the user level is carried out in a separate module and based on the output of the calculated user level, adaptive quizzes can be presented to the

individuals. The original poster's level of the question to the platform will be considered the question difficulty level. For example, if a beginner level member asks a question, that question's difficulty level will be marked as 'Beginner/Easy,' and once a beginner user is trying the quizzes, the platform will suggest the above-stated beginner questions first. However, the user is also not restricted from trying out any other difficulty-level questions. Since the platform accurately calculates levels using an algorithm, the platform will present unique and adaptive questions for individual knowledge levels.

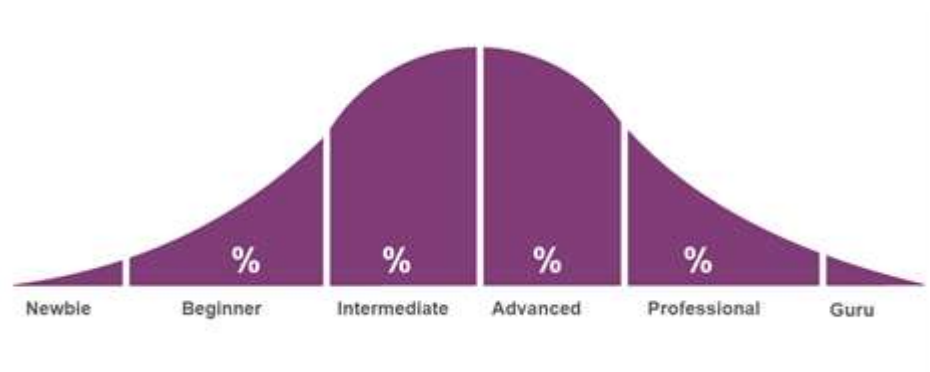


Figure 2-2: User level distribution using bell curve

2.1.2 Keyword Extraction in answers.

Keyword Extraction (KE) is the automated extraction of single or multiple-token phrases from a textual document that best expresses all critical aspects of its content and can be seen as the automated generation of a short document summary[21]. In the ProbExpert platform, those extracted keywords will be used to calculate the relevancy of the information. Since a significant focus in this research is to introduce a universal scoring method, keywords play a significant role in that prospect. The users of the platform are much more technical persons and therefore expecting an accurate output. To achieve this task, a keyword extraction method is used to minimize the redundancy and the effect of non-technical words in the similarity checking process. Following the above practice, the main goal is to identify the technical keywords present in answer to formulate an accurate answer.

The first part of the process is to make a pool of candidate keywords of the user and platform answer, which can be done with SciKit-Learn's *CountVectorizer* also, all the stop words will be removed. The use of the *n-gram-range* will change the size of the resulting candidates. Once all the candidates are selected, it will be added to the process of POS (Part-of-Speech) tagging with SpaCy that helps identify and tag each selected candidate's context. The application will only use nouns. Once the process is finished, the next step is keyword generation. We use the pre-trained hugging face model '*distilroberta-base*' and *AutoModel* and *AutoTokenizer* from Transformers to achieve this task. Once the candidates and the answer texts are tokenized and embedded. The last step of the process is to measure the distance using cosine similarity to measure the similarity between candidates and the original document, in this case, the answer, and get the desired number of keywords.

In summary, the Embedding-based keyword extraction method has been used to extract keywords. This exploits document embeddings and cosine similarity to identify candidate keywords. First, a document embedding is computed, then word n-grams of different sizes are generated, which are subsequently ranked along with their similarity to the embedding of the document.

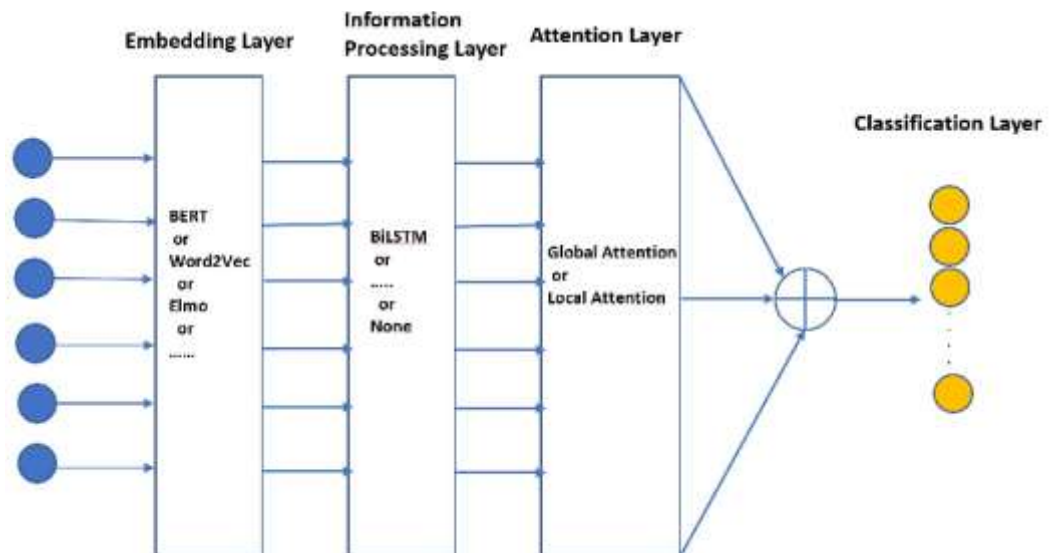


Figure 2-3: Keyword extraction using BERT

The below illustrations show how keyword extraction is achieved gradually within the system. The input text is the answer to the question, ‘What is a python module?’.

Input question	What is a python module?
Platform answer	The python module is a python object with arbitrarily named attributes that you can bind and reference. Simply, a module is a file consisting of Python code. A module can define functions, classes, and variables. A module can also include runnable code.
Model Initialization	<pre>model_name = "distilroberta-base" model = AutoModel.from_pretrained(model_name) tokenizer = AutoTokenizer.from_pretrained(model_name)</pre>
Initial Candidate Selection (Platform answer)	<pre>[] all_candidates[:10] ['arbitrarily', 'arbitrarily named', 'attributes', 'attributes bind', 'bind', 'bind reference', 'classes', 'classes variables', 'code', 'code module']</pre>

<p>Candidates after POS tagging with SpaCy</p>	<pre>[] candidates[:10] ['attributes', 'classes', 'code', 'file', 'functions', 'module', 'object', 'python code', 'python module', 'runnable code']</pre>	
<p>After generating the final keyword using the '<i>distilroberta-base</i>' model and distance measuring using cosine similarity. (Platform answer)</p>	<pre>[] keywords ['code', 'object', 'classes', 'module', 'variables', 'file', 'functions', 'runnable code', 'python module', 'python code']</pre>	
<p>User answer</p>	<p>Python modules are files containing Python code. This code can either be functions classes or variables. A Python module is a .py file containing executable code.</p>	

Candidates after POS tagging with SpaCy(user answer)	<pre>[12] candidates[:10] ['classes', 'code', 'executable code', 'file', 'files', 'functions', 'functions classes', 'module', 'modules', 'python code']</pre>	
After generating the final keyword using the 'distilroberta-base' model and distance measuring using cosine similarity. (user answer)	<pre>keywords ['module', 'variables', 'classes', 'modules', 'functions classes', 'files', 'functions', 'executable code', 'python modules', 'python code']</pre>	

Table 2-1: Stages of keyword generation

2.1.3 Summarization of the answers.

In order to construct the optimal scoring method, a summarization task will be carried out. The purpose of this is also to neglect the user's unique answering style. Summarizing the two answers (platform and user) using the same model helps bring both answers on to a common ground and improve the accuracy of the final scoring equation.

The preferred summarization model is Google's T5 which is trained with transfer learning and presents state-of-the-art results in summarization tasks. To carry out the methodology PyTorch and *AutoTokenizer*, *AutoModelWithLMHead* objects from the Transformer library will be necessary. Then tokenizing will be performed, and the tokenized data can be summarized by calling the T5 model's *model.generate* function. In this process, the inputs will be summarized to a maximum length of 50 words.

Input question	What is a python module
Platform answer	<p>A python module is a python object with arbitrarily named attributes that you can bind and reference.</p> <p>Simply, a module is a file consisting of Python code.</p> <p>A module can define functions, classes, and variables.</p> <p>A module can also include runnable code.</p>
Model Initialization	<pre>tokenizer = AutoTokenizer.from_pretrained('t5-base') model = AutoModelWithLMHead.from_pretrained('t5-base', return_dict=True)</pre>
Answer tokenizing into tensor vectors	<pre>inputs = tokenizer.encode('summarize: ' + sequence, return_tensors='pt', max_length=512, truncation=True)</pre> <p>inputs</p> <pre>tensor([[21605, 10, 20737, 6008, 19, 3, 9, 20737, 3735, 28, 17834, 1665, 128, 2650, 12978, 34, 25, 54, 3, 8610, 11, 2648, 5, 7383, 474, 6, 3, 9, 3, 102, 63, 189, 106, 6008, 19, 3, 9, 1042, 5688, 53, 13, 20737, 1081, 5, 71, 6008, 54, 8634, 3621, 6, 2287, 11, 11445, 5, 71, 6008, 54, 92, 568, 861, 29, 179, 1081, 5, 1]])</pre>
The summarized answer of the platform	<pre>summary = tokenizer.decode(summary_ids[0]) print(summary)</pre> <p><pad> a python module is a file consisting of Python code. It can define functions, classes and variables. a module can also include runnable code. (/)</p> <p>‘a python module is a file consisting of Python code. it can define functions, classes and variables. a module can also include runnable code.’</p>

User answer	<p>Python modules are files containing Python code.</p> <p>This code can either be functions classes or variables.</p> <p>A Python module is a .py file containing executable code.</p>
Answer tokenizing into tensor vectors (user answer)	<pre>inputs tensor([[21603, 10, 20737, 10561, 33, 2073, 3, 6443, 20737, 1081, 5, 100, 1081, 54, 893, 36, 3621, 2287, 42, 11445, 5, 71, 20737, 6008, 19, 3, 9, 3, 5, 102, 63, 1042, 3, 6443, 9362, 179, 1081, 5, 1]])</pre>
The summarized answer of the platform (user answer)	<pre>print(summary) <pad> a.py file containing executable code is a.py file containing Python code.</s></pre> <p>‘a.py file containing executable code is a.py file containing Python code.’</p>

Table 2-2: Answer summarization

2.1.4 Comparing answers

Another main objective of this research is to compare platform and user answers to maximize the scoring accuracy. Popular ways to achieve this are using TF-IDF or Word2Vec along with the cosine similarity. Although those are popular methods, the accuracy comes to question as a technique like TF-IDF was introduced in the early 2000s, and the NLP has progressed rapidly within the past five years. Our solution uses BERT-based transformers, specifically *sentence-transformers* library, that have the pre-trained model ‘*bert-base-nli-mean-tokens*.’

Once the user and the platform answer are retrieved from the database, the BERT sentence transformers model map answers to a 768-dimensional dense vector space, enabling tasks like clustering and semantic search. After tokenizing, embeddings, and performing mean pooling take attention mask into account for correct averaging and ignore the non-real tokens. After the cosine

similarity between the two vectors is completed, a similarity score between the two answers mentioned above can be retrieved. Behind the scenes, the calculation of the cosine similarity of the user question (u) and the platform question is illustrated below.

$$\text{similarity}(u, v) = \frac{u \bullet v}{\|u\| \|v\|} = \frac{\sum_{i=1}^n a_i b_i}{\sqrt{\sum_{i=1}^n u_i^2} \sqrt{\sum_{i=1}^n v_i^2}}$$

Figure 2-4: Cosine similarity formula

Input question	What is a python module
Platform answer	Python module is a python object with arbitrarily named attributes that you can bind and reference. Simply, a module is a file consisting of Python code. A module can define functions, classes, and variables. A module can also include runnable code.
User answer	Python modules are files containing Python code. This code can either be functions classes or variables. A Python module is a .py file containing executable code.
Model Initialization	<pre> sentences = [platform_answer, user_answer] model_name = "sentence-transformers/bert-base-nli-mean-tokens" </pre>

Tokenization	<pre>[] tokens = {'input_ids': [], 'attention_mask': []} # For sentence in sentences: new_tokens = tokenizer.encode_plus(sentence, max_length=128, truncation=True, padding='max_length', return_tensors='pt') tokens['input_ids'].append(new_tokens['input_ids']) tokens['attention_mask'].append(new_tokens['attention_mask']) [] tokens['input_ids'] = torch.stack(tokens['input_ids']) tokens['attention_mask'] = torch.stack(tokens['attention_mask']) [] tokens['input_ids'].shape torch.Size([2, 128])</pre>	
Attention mask embedding	<pre>embeddings = outputs.last_hidden_state embeddings.shape torch.Size([2, 128, 768]) attention = tokens['attention_mask'] attention.shape torch.Size([2, 128]) mask = attention.unsqueeze(-1).expand(embeddings.shape).float() mask_embeddings = embeddings * mask mask_embeddings.shape torch.Size([2, 128, 768]) summed = torch.sum(mask_embeddings, 1) summed.shape torch.Size([2, 768]) counts = torch.clamp(mask.sum(1), min=1e-9) counts.shape</pre>	
Mean Pooling	<pre>[] mean_pooled = summed / counts mean_pooled.shape torch.Size([2, 768]) [] mean_pooled tensor([[0.3447, 0.2290, -0.0283, ..., -0.1281, -1.0052, 0.4548], [0.1317, 0.2670, 0.1589, ..., -0.2130, -1.1523, 0.5539]], grad_fn=<DivBackward0>)</pre>	

Cosine similarity calculation	<pre>[] mean_pooled = mean_pooled.detach().numpy() cosine_similarity([mean_pooled[0]], mean_pooled[1:]) array([[0.957978]], dtype=float32)</pre>
--	---

Table 2-3: Comparing answers

A significant weight is put on answer comparison like the above-stated keyword extraction, and the summarization outputs also act as an input to this comparing answer implementation because we measure the similarity of the through this. As a result, the raw answers of both the platform and the user show a similarity score of 0.958.

2.1.5 Scoring method

Once the keyword extraction model extracted the relevant keywords and the summarization model summarized both the user and the platform answers, the resulted in two outputs will be recalculated for the similarity using the answer comparing model, thus giving three types of similarity scores. Below are the expected three scenarios.

- The cosine similarity score between the platform's optimal answer and the user's answer. (C_o)
- The cosine similarity score of the summarized user answer and the platform's optimal answer. (C_s)
- Cosine similarity of user answer's extracted keywords and platform's answer's extracted keywords. (C_k)

Since the scorer method is theoretically involved with three different BERT models (*distilroberta-base*, *bert-base-nli-mean-tokens*, *t5-base*) for final answer evaluation, it helps to eliminate the dependency of a single model. Also, it eliminates any single BERT transformer model-related inconsistencies[27]. Once all the similarity scores are obtained, the mean score will be presented as

the final score. If the calculated cosine similarity score can be represented in C and the final score in S , the formula follows.

$$S = \frac{C_o + C_s + C_k}{3}$$

To demonstrate the above formula, we calculated the cosine score of each based on the example question that has been used so far. The results are as follows:

Cosine similarity Component	Cosine similarity score
The cosine similarity score between the platform's optimal answer and the user's answer. (C_o)	0.958 <code>array([[0.957978]], dtype=float32)</code>
The cosine similarity score of the summarized user answer and the platform's optimal answer. (C_s)	0.891 <code>array([[0.8913338]], dtype=float32)</code>
Cosine similarity of user answer's extracted keywords and platform's answer's extracted keywords. (C_k)	0.979 <code>array([[0.97871494]], dtype=float32)</code>

Table 2-4: Scoring model breakdown

By combining the results into the constructed formula, the platform can generate a collective, unbiased score for a quiz.

$$S = \frac{0.958 + 0.891 + 0.979}{3}$$

Therefore, the final score will be 0.943, which then can be converted into a percentage when presenting to the user, thus resulting in a 94.3% as the final score.

2.1.6 Web Scrapping for non-existing questions and answers.

ProbExpert platform is still at an early age; therefore, situations may arise where users can not find the desired quiz questions. To address the scenario, the platform is equipped with a web scraper developed with *BeautifulSoup* and *lxml*. With the scrapper's help, users will get questions and relevant answers from the internet resources such as StackOverflow. Hence this process involves an external source, and the platform can not vouch for the accuracy of the answer. Therefore, the similarity checking and the score assignment will not be carried out. Once ProbExpert is matured, this functionality will rarely be needed.

2.1.7 Pre-processing of the data

A large amount of raw data was used to generate information in ProbExpert. As the first task, the data must be processed by removing unnecessary data and converting the data into a numerical representation. There are three steps in data processing.

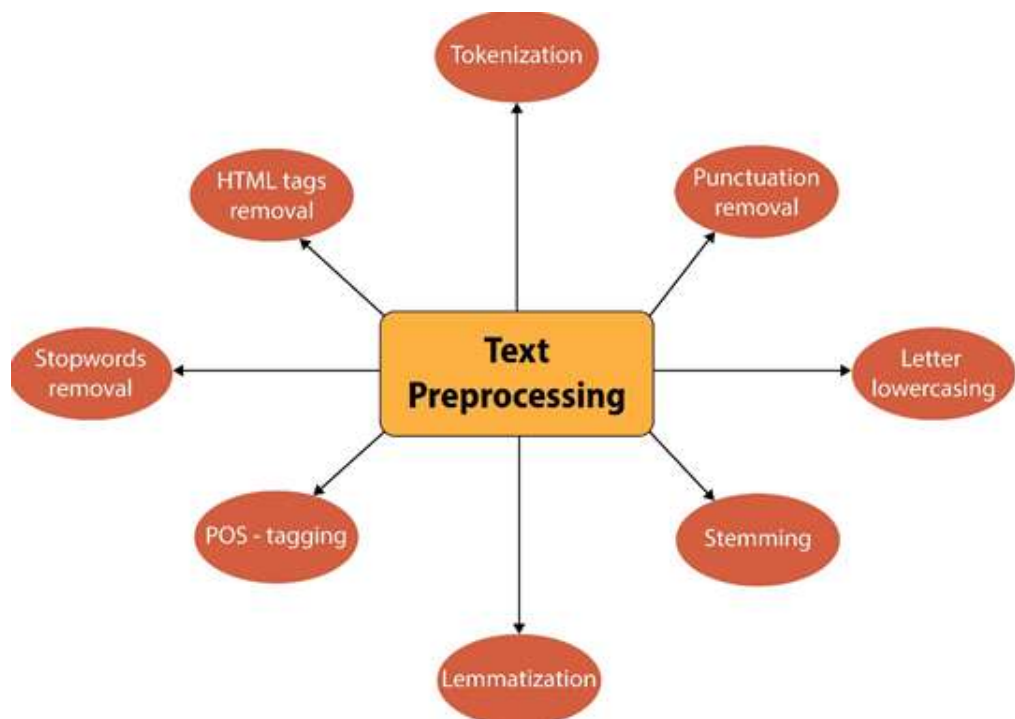


Figure 2-5: Text preprocessing overview

Tokenization: In this process, the text is first tokenized into small individual tokens such as words, punctuation. This process is done by the implementation of rules specific to each language. Based on the specified pattern, the strings are broken into tokens using regular expressions, The patterns used in this work remove the punctuations.



Figure 2-6: Tokenization process

Stop word removal: The stop words are a group of often used words in the language. Like in English, having several stop words such as "the," "a," "is," "are,." The perception of using these kinds of stop words is that removing common informative words from the text could focus more on the crucial words.

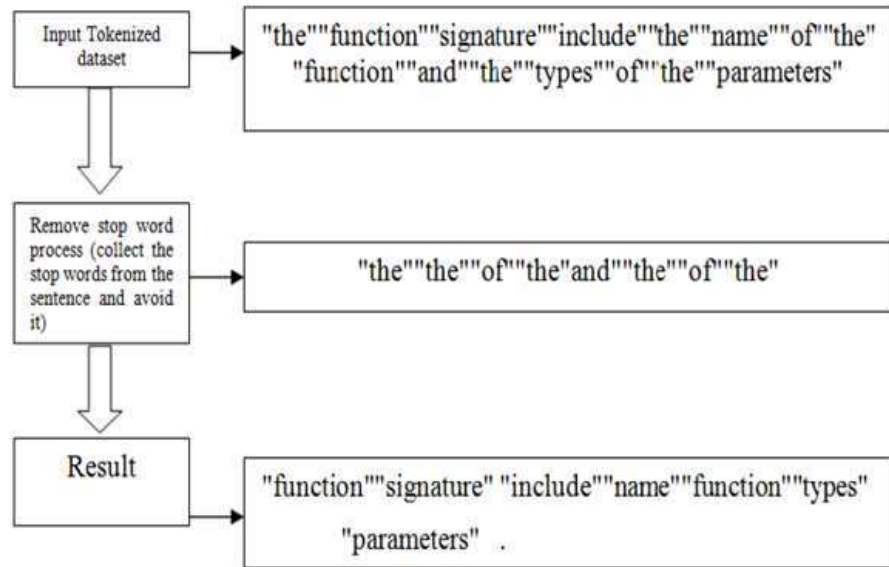


Figure 2-7: Stop word removal

Text Embedding using transformers: The text form of a sentence cannot be utilized in natural language processing to verify the similarity of a sentence to another sentence (questions and answers according to this research component). Sentences must be translated into numerical form. Text data is converted into numerical vector representation to the algorithm to generalize the data and perform further steps.

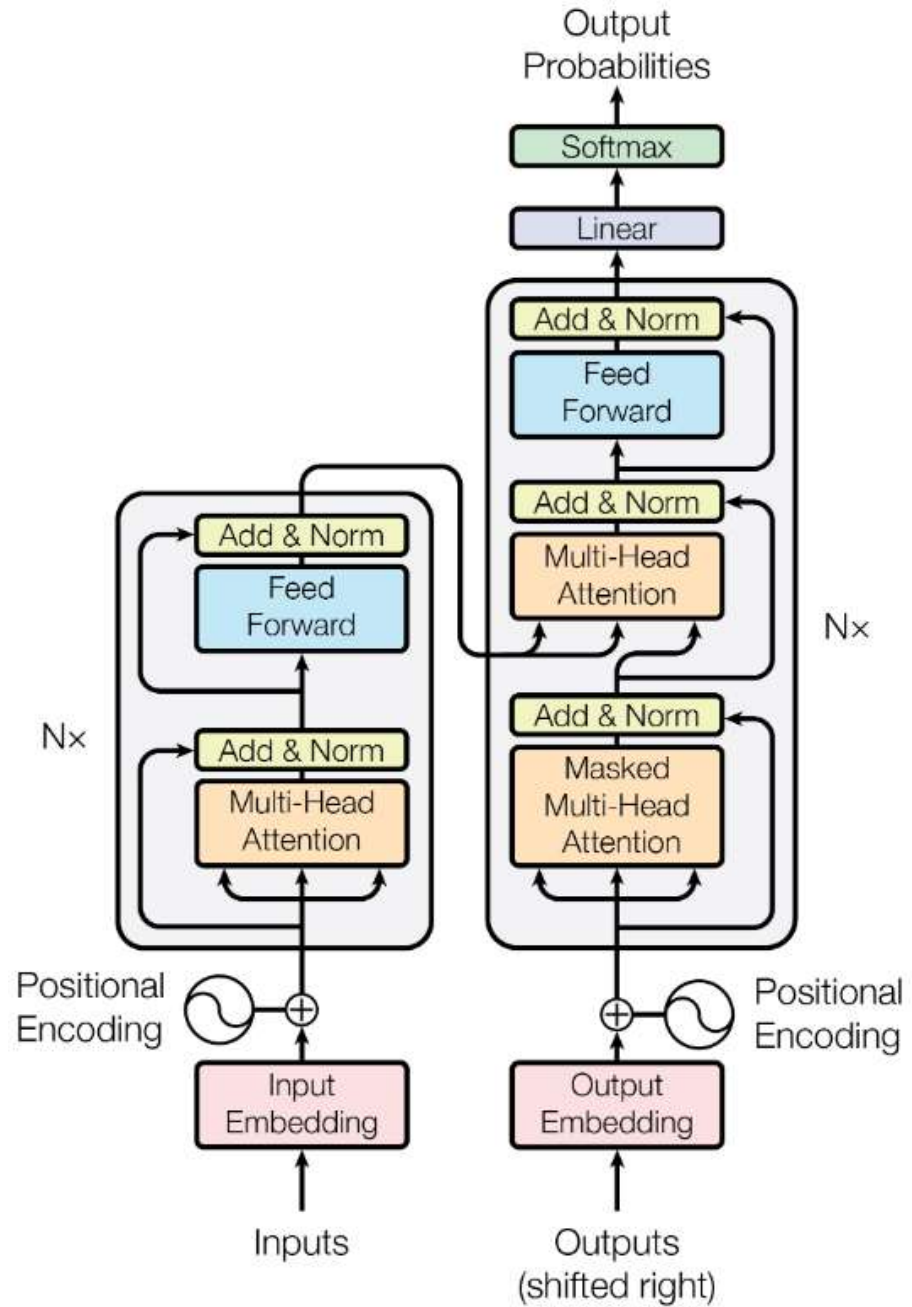


Figure 2-8: Transformer based word embedding

2.1.8 Models

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

- *DistilRoBERTa base model*: The RoBERTa-base model has been distilled into this model. It goes through the same training as DistilBERT. 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.
- *Google's T5-base*: Transfer learning has emerged as a powerful technique in natural language processing, where a model is first pre-trained on a data-rich task before being fine-tuned on a downstream task (NLP). Transfer learning's effectiveness has spawned a plethora of approaches, methodologies, and practices. T5-base model is created using the above principle and thus resulting in state-of-the-art results on many benchmarks covering summarization, question answering, text classification, and more. The monthly download average states close to 2 million, therefore indicating the effectiveness and the popularity.
- *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.

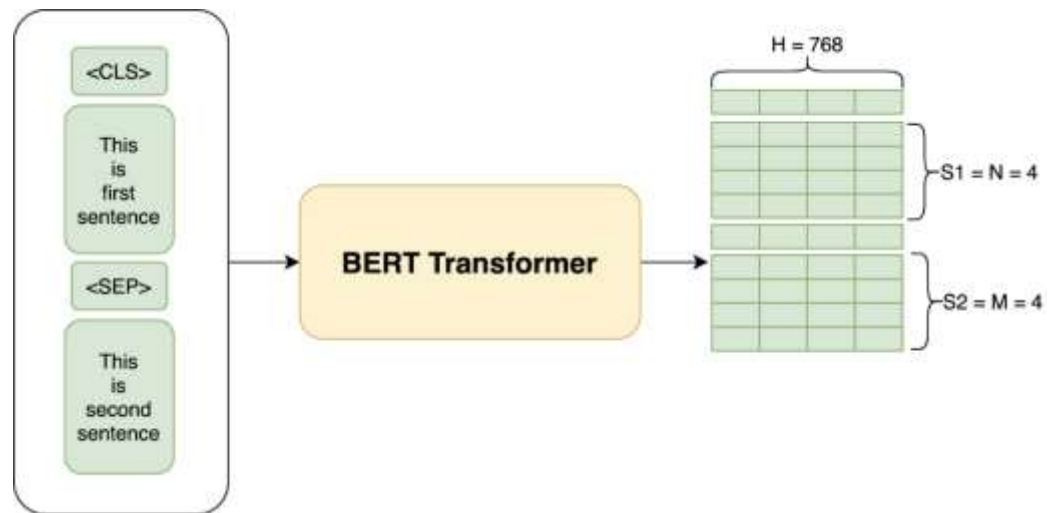


Figure 2-9: High-level functionality of a BERT Transformer

2.2 Commercial Aspect of the Product

Introducing ProbExpert into the market is a fundamental goal in this project. The below figure illustrates the marketing plan and the strategy we made for ProbExpert commercialization purpose. Our business goal is to achieve commercial success within one year. By then the userbase is expected to grow and the platform is advertised and promoted over the internet.

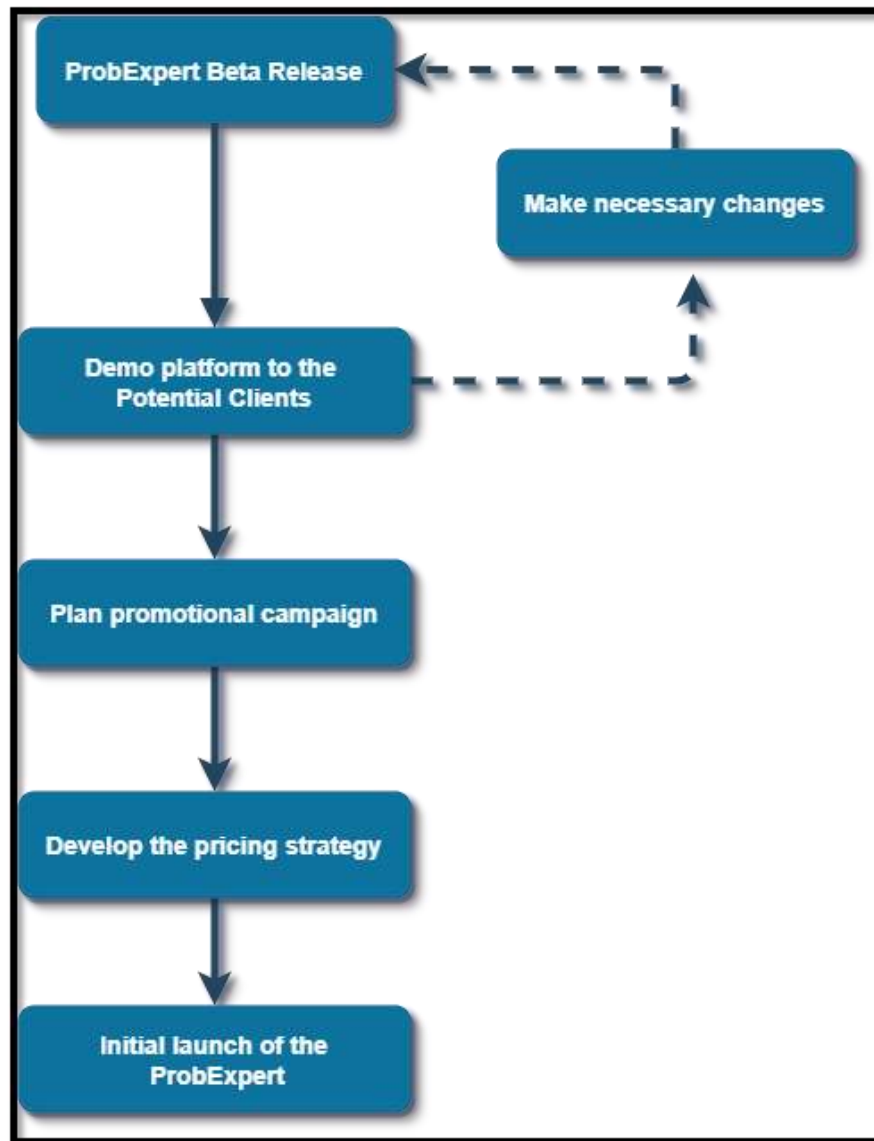


Figure 2-10: ProbExpert's Marketing Strategy

ProbExpert has the potential of commercially succeeding around below stated aspects.

- Subscription-based usage is a popular business model in present-day service providers. ProbExpert can adopt the above business model in two separate user groups. As stated previously, theoretical knowledge is equally essential to a candidate in general, but companies focus on evaluating the theoretical

knowledge when applying for a profession. With this use case in mind, a premium subscription can be introduced in two categories, individual and business subscription packages.

- *Premium Subscription for individual users:* In the quiz section, regular users will only have access to complete 20 questions per day; therefore, users can purchase a subscription monthly or annually for unlimited access to the question set and answers.
 - *Premium subscription for business users:* Businesses can get the best of our platform by using it as an internal resource for employees. Existing employees can use the platform to enhance and share their knowledge as well as the company can evaluate them. Another use case for an organization is the ability to use to evaluate new recruits on theoretical knowledge as many interview processes include a theoretical session.
- Another commercialization opportunity of ProbExpert is to become an optimal platform for advertisements. However, only a few niches related to Information technology will be allowed to publish on the platform. Allowed advertisements can be categorized into two sections as recognized here.
- *Job Openings:* Organizations can display their available vacancies in the related section for jobs to attract brilliant minds to the company. Interested parties then can contact and get along with the company process. This service can be provided as a pay-as-you-go service or a fixed monthly or annual amount to display a limited number of advertisements according to the selected package.
 - *Product/Service Advertisements:* Since the platform is populated with technical people of the IT industry, companies can advertise their tech-related products or services. i.e., Laptops, Hosting, and domains, cloud storage. These custom ads can potentially bring value to both users and the platform as well.

3 TESTING & IMPLEMENTATION

3.1 Testing

The goal of testing is to find and fix flaws in the system that has been developed. A tested programming system can be identified as a verified and approved system, and the testing phase is necessary to achieve the system. It is a method of verification and validation. The system is subjected to unit testing, system testing, and acceptance testing, which will help determine whether the objectives can be met or not.

a) Unit testing

Individual programs, subroutines, procedures, and, in general, each unit in a system are all subjected to unit testing. It is a method of coordinating the various aspects of testing. Unit testing should be done separately for each module. This serves two purposes: it improves unit performance without breaking it and lowers the costs of repairing failures. This is beneficial because it can be repaired appropriately when a failure is discovered. After all, the testing is done separately. It makes debugging tasks easier. The system began as a series of small programs that were later integrated into the final phase.

b) Testing for integration

Individual parts are combined and tested as a whole in Integration Testing. The goal of performing integration testing is to ensure that the evaluation of two or more components produces a result that meets the functional requirement. After unit testing and before validation testing, this is done. Module testing in this environment always began at the top of the programming hierarchy and progressed downward.

Because testing begins early in the implementation process, failures can be detected sooner rather than later in the development cycle. It's simple to integrate because it's simple to test in a development environment. Integration

testing is more efficient than end-to-end testing. Isolating failures is reliable and straightforward. The developer and tester attitudes are also required for integration testing. Integration testing is required because ProbExpert was built using multiple programming languages and technologies. This ensures that the overall system functions appropriately. In this step, we tested backend to frontend integration.

c) System Testing

This evaluates the entire system or developed program in relation to its original goals. System testing is an attempt to figure out why a system is not meeting its goals. It validates not only the system's design and development but also the requirements of the users. The outputs of integration testing were passed as inputs in system testing. It is capable of detecting failures in both integrated units and the entire system. The final result of this testing method is to observe the system's behavior. In general, system testing is carried out by a separate testing team from the implementation team and is responsible for ensuring the system's quality. Both functional and non-functional testing are included.

d) Acceptance Testing

Acceptance testing is the process of comparing a system's initial requirements to the end users' current requirements. This is usually done by the customer or the end-user. Before handing over the system to the user, the developer will conduct user testing.

e) Regression Testing

Developers must change or modify functionality during the process; updates may result in unexpected behaviors, which can significantly impact. Regression testing is typically used to ensure that a change or addition has not affected any existing functions. Its goal is also to find bugs and errors that may have been

accidentally introduced into the existing solution and ensure that previously erased bugs are still active. Many functional testing tools are available for regression testing to continue their work.

f) API Testing

Because the ProbExpert API was built as a serverless backend, API testing is required to ensure that all endpoints are functioning correctly. API testing has examined security, availability, and response time. APIs of the ProbExpert platform have been tested using manually written test cases.

3.2 Implementation

The section covers all aspects of the system's implementation. Interfaces and coding are used to improve the system's coordination and development. To keep the system error-free, several tests should be performed at each stage.

- Frontend Development

- ReactJS
- NextJS Server-Side Rendering

The frontend of the ProbExpert quiz section was built with Next.js on top of React.js. The frontend components and stylings were created using the Material-Ui component library. The main reason for using Next.js to build the frontend is that it comes with server-side rendering (SSR) built-in.

- Backend Development

- NodeJS
- Python Server/Flask

To communicate with the ProbExpert frontend, the quizzes backend was built as a RESTful API service. The backend was built using the Express.js library on top of Node.js, and to execute ML tasks, Flask was

also used. Because Express.js has such a sophisticated routing mechanism, it dramatically reduces development time.

- Database

- MongoDB (NoSQL)

Because the quiz model contains a large number of data points, the majority of which are not accessible to all users, a database with a strict schema is more difficult to maintain. As a result, the MongoDB Atlas database has been chosen as the central database.

- Environment Handling

- NPM – Node package manager

npm is the package manager for the Node JavaScript framework. It installs modules so that node can find them and handles dependency conflicts intelligently. It can be designed to work in a variety of situations. It is primarily used to release, locate, download, and create node programs. In a web application, npm will be used to manage modules.

- PIP – Python package manager

PIP is a Python package management system that simplifies the installation and management of software. It connects to the Python Package Index, which is an online repository of both free and paid Python packages. On the web server, the Python Package Manager (PIP) will be used to manage Python packages.

- Deployment

Since the ProbExpert frontend was developed using Next.js, it has been deployed on their native Vercel's cloud platform. Vercel is a deployment and collaboration platform for deploy client-side applications and web services. These servers scale automatically according to the traffic, and they have an inbuilt caching system for all the applications.

Therefore, Vercel is the optimal solution to host the front end of the ProbExpert platform.

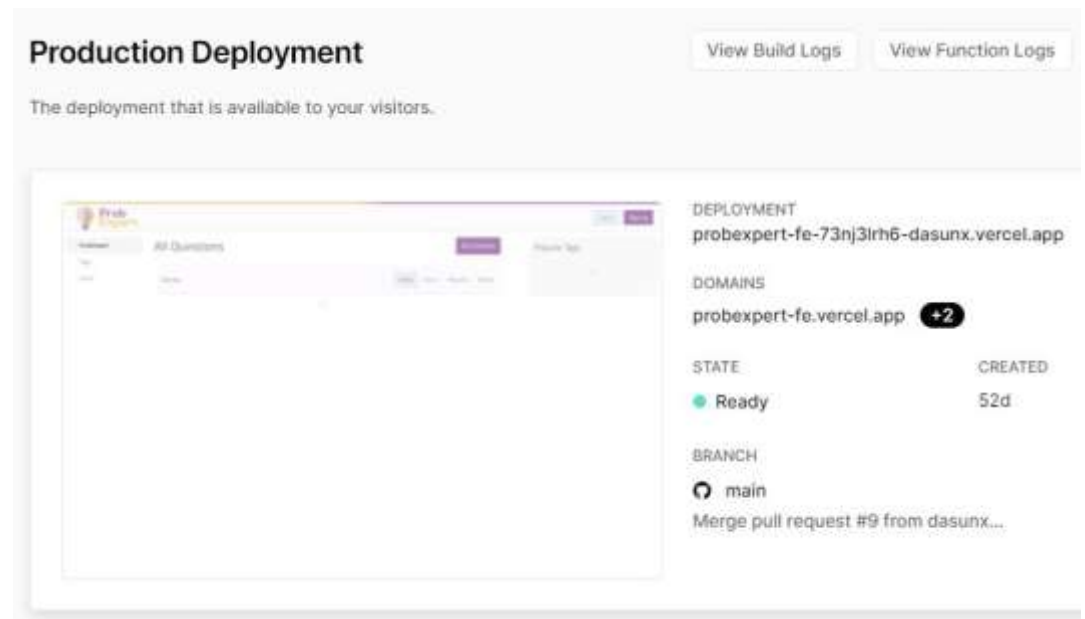


Figure 0-1: FrontEnd deployment

4 RESULTS & DISCUSSION

4.1 Results

Modern advancements in NLP are outperforming the popular implementations used in the previous decade. Especially newly introduced transformers and python libraries are advanced and have the capability to perform various NLP tasks with ease.

The ProbExpert platform has been able to produce accurate results using above mentioned novel technologies in the implementations, such as popular pre-trained BERT models to achieve various tasks.

Using these technologies, the system is equipped with several features to accurately measure the answers to the quiz questions. To achieve this task, implementations of keywords extraction, Summarization, and answer comparing mechanisms were combined as discussed in the *section 2.1*. We introduced a scorer model that does not rely on a single model but on three different models that are trained with hundreds of millions of real-world data to minimize the marking errors. The users are presented with questions about their desired subject, and the web scrapper has resulted in a seamless experience in quiz taking with the adaptiveness to each individual's knowledge/expert level.

4.2 Research Findings

Through completing this research, the introduced platform ProbExpert has addressed all the raised questions of the research problem since the research area is novel as no other platform had previously attempted to utilize their question database into a platform where users are able to test their theoretical knowledge with the structured type questions based on the questions of the real-world users.

We have found through the research that the actual gap of not having a proper platform to evaluate theoretical knowledge and most of the platforms only focus on polishing coding skills. Therefore, users are in dire need of a platform that offers the facility to test their theory knowledge before an important event such as an interview or an exam.

Also, we have noticed the difficulty to introduce a scoring method on theory-based text answers as the semantic input needs to be well analyzed. As discussed earlier, inputs can be varied according to each individual writing style and can not produce a straightforward score like creating a test case-based evaluation.

4.3 Discussion

The main objectives were to formulate structured type questions for knowledge checking using existing answered questions of the platform based on the user level and then introduce an unbiased scoring model. Moreover, by the evidence mentioned in previous chapters, we were successfully able to achieve the objectives.

The system was put through its paces with fictitious data. These figures were used to confirm and verify the formation as well as the connections between the various components. The system was then put through its paces in a real-world setting. Dummy data was used to verify the system's functionality and components in a real-world scenario. This process is repeated several times to ensure that the proposed system is feasible to implement. The system's final outputs confirmed that the functions are accurate in a real-world setting as well.

4.4 Summary of the Student Contribution

Personal	Functionality	Description
S.K.C.W.K.M.R.T.S.B. Marapana	ProbExpert Platform	<ul style="list-style-type: none"> • System design • Database design • Backend structure design • API structure design • Overall UI developments • Common UI component developments
	Quiz and scoring generation	<ul style="list-style-type: none"> • Design user interface • Capture data • Create Training model • Generate user scores of the quizzes • Generate adaptive quizzes • Web scraping

Table 0-1: Student contribution

5 CONCLUSION

As developers or individuals who are working in the information technology field, it is no surprise that everyone knows the importance of theoretical knowledge of programming concepts. Complex theoretical applications can be found in the everyday life of a developer. Most people can code, but what differentiates between a good and a better developer is the ability to integrate the learned theories into a code to make programs work efficiently.

Through this research, we have attempted to introduce a platform with various achievements. Among those, a section is dedicated to improving the programming knowledge and turn the user from an average developer to a better developer with the help of machine learning and natural language processing. We were able to present user-adaptive questions based on their knowledge level using the platform's asked questions and then allowing users to provide answers to evaluate their knowledge level by introducing a novel scoring model.

Future research will be involved in further optimizations on the scoring model and provide MCQ-type questions. ProbExpert is not a traditional Q&A website. While it helps achieve the primary cause, another out-of-the-box functionality is that the platform can act as a medium to improve individuals' programming knowledge. Being such a platform leads to having an enormous dataset of day-to-day programmer's questions and received reliable, optimized answers, which can be easily turned into a platform for theoretical knowledge checking. In this scenario, structured-type online quizzes are considered an effective method[4] and have proven a positive influence on academics. The questions are presented based on the user level, therefore enabling the adaptiveness to everyone for increased engagement and motivation. Presented questions will be generated by the platform's questions. Users can type the answer for each question, and they will receive an unbiased score to assess their current knowledge level.

6 REFERENCES

- [1] Salas-Morera, Lorenzo & Arauzo-Azofra, Antonio & Garcia-Hernandez, Laura. (2012). Analysis of online quizzes as a teaching and assessment tool. *Journal of Technology and Science Education*. 2. 39-45. 10.3926/jotse.30.
- [2] Roediger, Henry & Putnam, Adam & Sumeracki, Megan. (2011). Ten Benefits of Testing and Their Applications to Educational Practice. 10.1016/B978-0-12-387691-1.00001-6.
- [3] Lewis, PhD, D., Trail, MS, T., Srinivasan, MEd., MPH, S., Lee, PhD, S.J. & Lopez, MEd, S. (2010). Knowledge check questions: Best practices for use of this instructional strategy. In J. Herrington & C. Montgomerie
- [4] (Eds.), *Proceedings of ED-MEDIA 2010--World Conference on Educational Multimedia, Hypermedia & Telecommunications* (pp. 2783-2788). Toronto, Canada: Association for the Advancement of Computing in Education (AACE). Retrieved February 26, 2021 from <https://www.learntechlib.org/primary/p/35034/>.
- [5] Ross, B., Chase, AM., Robbie, D. *et al.* Adaptive quizzes to increase motivation, engagement and learning outcomes in a first year accounting unit. *Int J Educ Technol High Educ* **15**, 30 (2018). <https://doi.org/10.1186/s41239-018-0113-2>.
- [6] Hsiao, I.-H., Sosnovsky, S. and Brusilovsky, P. (2010), Guiding students to the right questions: adaptive navigation support in an E-Learning system for Java programming. *Journal of Computer Assisted Learning*, 26: 270-283. <https://doi.org/10.1111/j.1365-2729.2010.00365.x>
- [7] Anatol, T., and S. Hariharan. "Reliability of the evaluation of students' answers to essay-type questions." *West indian medical journal* 58.1 (2009).
- [8] Baltadzhieva, Antoaneta, and Grzegorz Chrupała. "Predicting the quality of questions on stackoverflow." *Proceedings of the international conference recent advances in natural language processing*. 2015.
- [9] L. Mamykina, B. Manoim, M. Mittal, G Hripcsak, and B. Hartmann, "Design Lessons from the Fastest Q&A Site in the West," in Proceedings of the 2011 annual conference on Human factors in computing systems, New York, NY, USA, 2011, pp. 2857-2866.

- [10] S. M. Nasehi, J. Sillito, F. Maurer and C. Burns, "What makes a good code example?: A study of programming Q&A in StackOverflow," 2012 28th IEEE International Conference on Software Maintenance (ICSM), Trento, Italy, 2012, pp. 25-34, doi: 10.1109/ICSM.2012.6405249.
- [11] P. Sitikhu, K. Pahi, P. Thapa and S. Shakya, "A Comparison of Semantic Similarity Methods for Maximum Human Interpretability," *2019 Artificial Intelligence for Transforming Business and Society (AITB)*, Kathmandu, Nepal, 2019, pp. 1-4, doi: 10.1109/AITB48515.2019.8947433.
- [12] M. Tang, P. Gandhi, M. A. Kabir, C. Zou, J. Blakey, and X. Luo, "Progress notes classification and keyword extraction using attention based deep learning models with BERT," *arXiv*, 2019.
- [13] <https://wiki.pathmind.com/bagofwords-tf-idf#:~:text=Each%20word's%20TF%20DIDF%20relevance,also%20adds%20up%20to%20one.&text=The%20main%20difference%20is%20that,content%20and%20subset%20of%20content>.
- [14] X. Jin, S. Zhang and J. Liu, "Word Semantic Similarity Calculation Based on Word2vec," 2018 International Conference on Control, Automation and Information Sciences (ICCAIS), Hangzhou, China, 2018, pp. 12-16, doi: 10.1109/ICCAIS.2018.8570612.
- [15] P. Yuan, A. Du and C. Wang, "Using Word2vec to Match Knowledge Points and Test Questions: A Case Study," 2020 IEEE 2nd International Conference on Computer Science and Educational Informatization (CSEI), Xinxiang, China, 2020, pp. 272-276, doi: 10.1109/CSEI50228.2020.9142504.
- [16] N. R. Ramadhanti and S. Mariyah, "Document Similarity Detection Using Indonesian Language Word2vec Model," 2019 3rd International Conference on Informatics and Computational Sciences (ICICoS), Semarang, Indonesia, 2019, pp. 1-6, doi: 10.1109/ICICoS48119.2019.8982432.
- [17] Palmer, E.J., Devitt, P.G. Assessment of higher order cognitive skills in undergraduate education: modified essay or multiple choice questions? Research paper. *BMC Med Educ* **7**, 49 (2007). <https://doi.org/10.1186/1472-6920-7-49>
- [18] I. N. Bandeira, T. V. Machado, V. F. Dullens and E. D. Canedo, "Competitive programming: A teaching methodology analysis applied to first-year programming

- classes," 2019 IEEE Frontiers in Education Conference (FIE), Covington, KY, USA, 2019, pp. 1-8, doi: 10.1109/FIE43999.2019.9028518.
- [19] R. R. A. M. P. Jayawardena, G. A. D. Thiwanthi, P. S. Suriyaarachchi, K. I. Withana and C. Jayawardena, "Automated Exam Paper Marking System for Structured Questions and Block Diagrams," 2018 IEEE International Conference on Information and Automation for Sustainability (ICIAfS), Colombo, Sri Lanka, 2018, pp. 1-5, doi: 10.1109/ICIAFS.2018.8913351.
- [20] R. Siddiqi and C. Harrison, "A systematic approach to the automated marking of short-answer questions," 2008 IEEE International Multitopic Conference, Karachi, Pakistan, 2008, pp. 329-332, doi: 10.1109/INMIC.2008.4777758.
- [21] M. Keikha, J. H. Park, and W. B. Croft, "Evaluating answer passages using summarization measures," 2014, doi: 10.1145/2600428.2609485.
- [22] L. Averell and A. Heathcote, "The form of the forgetting curve and the fate of memories," *J. Math. Psychol.*, vol. 55, no. 1, pp. 25–35, 2011, doi: 10.1016/j.jmp.2010.08.009.
- [23] H. L. Roediger, A. L. Putnam, and M. A. Smith, *Ten Benefits of Testing and Their Applications to Educational Practice*, vol. 55. 2011.
- [24] J. Piskorski, N. Stefanovitch, G. Jacquet, and A. Podavini, "Exploring Linguistically-Lightweight Keyword Extraction Techniques for Indexing News Articles in a Multilingual Set-up," *Proc. EACL Hackashop News Media Content Anal. Autom. Rep. Gener.*, pp. 35–44, 2021.
- [25] S. Desai and G. Durrett, "Calibration of Pre-trained Transformers," pp. 295–302, 2020, doi: 10.18653/v1/2020.emnlp-main.21.
- [26] T. Wolf *et al.*, "Transformers: State-of-the-Art Natural Language Processing" pp. 38–45, 2020, doi: 10.18653/v1/2020.emnlp-demos.6.
- [27] M. Grootendorst, "Keyword Extraction with BERT," 2020. .
- [28] D. Gunawan, C. A. Sembiring, and M. A. Budiman, "The Implementation of Cosine Similarity to Calculate Text Relevance between Two Documents," *J. Phys. Conf. Ser.*, vol. 978, no. 1, 2018, doi: 10.1088/1742-6596/978/1/012120.
- [29] B. Li and L. Han, "Distance weighted cosine similarity measure for text classification," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 8206 LNCS, pp. 611–618, 2013, doi: 10.1007/978-3-642-41278-3_74.
- [30] A. Chernyavskiy, D. Ilvovsky, and P. Nakov, "Transformers: 'The End of History' for NLP?," 2021