ISSN: 0067-2904

# Answers Generation Based on English Textual Analyzer (AGETA)

**Hiba Amjed Mohamed*, Abeer Khaled Al-Mashhsdany**
*Computer Science, College of Science, Al-Nahrain University, Baghdad, Iraq*

Received: 20/1/2025 Accepted: 11/5/2025 Published: 30/5/2026

**Abstract**

Accurate and contextually appropriate answer generation is increasingly important for applications in virtual assistants, educational tools, search engines, and so on, for answering any question about information found across electronic libraries. This research, "Answers Generation based on English Textual Analyzer (AGETA)," hopes to generate the correct answer as a complete comprehension sentence. It receives a passage with a related list of questions in English as unstructured typed texts. It performs the English textual analysis process using a series of natural language processing (NLP) techniques, followed by a hybrid method that combines extractive techniques with linguistic analysis to build an effective answer generator, as a set of its main applied techniques: tokenization, part-of-speech tagging, cosine similarity, T5 model, and then applying grammar-checking mechanism and English syntactic rules. AGETA was tested on questions related to different passages, with performance measured using two types of accuracy measurements. The first type is human performance, which achieved 92% for short answers and 96% for expanded answers. The second type is Sentence Transformers (BERT-based models), which achieved 90%. This indicates that the generated answers exhibit considerable unity with the corresponding ground truth answers. The proposed approach has potential applications in education, research, and customer support by enhancing the accessibility and relevance of textual information.

**Keywords:** Textual analysis, Keyword Extraction, part-of-speech tagging, deep learning model, English linguistic rules.

# توليد الاجابات استنادا" الى محلل النصوص الانكليزية

**هبه امجد محمد*, عبير خالد المشهداني**
علوم الحاسوب, كلية العلوم, جامعة النهرين, بغداد, العراق

**الخلاصة**

إن توليد الإجابات الدقيقة والمناسبة للسياق أمر مهم بشكل متزايد للتطبيقات في المساعدين الافتراضيين والأدوات التعليمية ومحركات البحث وما إلى ذلك للإجابة على أي سؤال حول المعلومات الموجودة على المكتبات الالكترونية. يأمل هذا البحث "توليد الإجابات بناءً على محلل النصوص الإنجليزية "(AGETA) في توليد الإجابة الصحيحة كجملة فهم كاملة. يتلقى مقطعًا مع قائمة الأسئلة الخاصة به كنصوص مكتوبة باللغة

*Email: heba.msco23@ced.nahrainuniv.edu.iq

الإنجليزية غير مهيكلة. يقوم بعملية تحليل النص الإنجليزي باستخدام سلسلة من تقنيات معالجة اللغة الطبيعية (NLP)، تليها طريقة هجينة تجمع بين تقنيات الاستخراج والتحليل اللغوي لبناء مولد إجابة فعال. كمجموعة من تقنياته التطبيقية الرئيسية وهي التجزئة، ووضع علامات على أجزاء الكلام، وتشابه Cosine، ونموذج T5، ثم تطبيق آلية التحقق من القواعد النحوية وقواعد النحو الإنجليزية. تم اختبار AGETA على أسئلة تتعلق بمقاطع مختلفة، مع قياس الأداء باستخدام نوعين من قياس الدقة، النوع الأول هو الأداء البشري، والذي حقق 93٪. النوع الثاني هو محولات الجملة (BERT-based models ) والتي حققت نسبة 90%. مما يشير إلى أن الإجابات الناتجة تظهر درجة كبيرة من الوحدة مع إجابات الحقيقة الأساسية المقابلة. النهج المقترح له تطبيقات محتملة في التعليم والبحث ودعم العملاء من خلال تعزيز إمكانية الوصول إلى المعلومات النصية وأهميتها.

## 1. Introduction

Generating answers from an English textual analysis has become crucial, especially given the vast amount of online information. Accurate and contextually appropriate answer generation is increasingly important for applications in virtual assistants, educational tools, and search engines [1]. Textual analysis provides methods for describing and interpreting the characteristics of texts. When textual analysis is performed on a text, an educated guess is made in some interpretations that are likely made of that text. So, textual analysis is the first step in developing an answer-generation method [2]. In the textual analysis process, it is important to improve the acquisition and selection of texts to be studied and determine the appropriate approach for analyzing them. There are many approaches: morphological and syntactic analysis, interaction analysis, content and semantic analysis, machine learning techniques, discourse analysis, rhetorical criticism, narrative analysis, and performance studies [3], [4]. Common text analysis applications include question answering, keyword extraction, information retrieval, text generation, document clustering, and document filtering [5]. As a preprocessing for those applications, some operations are needed, such as extracting features, representing documents, and signature creation. In most text analysis applications, the preprocessing operations are the same [6]. Text generation is one of the broad and distinct fields in natural languages due to its importance in enabling machines to express themselves like humans in many fields, such as translation, conversation, summarization, and answer generation [7]. The process of generating texts is done through several techniques, some of which use neural networks such as (BERT) and some of which use statistical analysis (n_gram), and others depend on analyzing linguistic information and retrieving information, which uses several linguistic techniques such as: named entity recognition (NER), part of speech-language (POS), syntactic parsing, and semantic analysis [8].

Generating answers is a significant challenge in natural language processing and a vital part of text generation. This process often faces difficulties with coherence and query ambiguity. While there has been considerable progress in natural language processing, particularly in text analysis, there remains a significant gap in the ability to generate long, coherent responses from complex textual data [9].

The authors of [10] (2021) introduced a system integrating question-answering and text-generation tasks to create question-and-answer pairs for multi-paragraph documents. The authors employed TF-IDF similarity and deep learning models, specifically BERT with SHAREDNORM. They utilized the SQUAD dataset derived from Wikipedia articles and the NEWSQA dataset from CNN News. The system achieved an exact match (EM) score of 72% and an F1 score of 80%.

The authors of [11] (2022) presented a semantic question-answering (QA) system for e-learning environments. The system utilized linguistic resources to understand and retrieve relevant answers effectively. The researchers processed the questions using Natural Language Processing (NLP) tools and semantic resources such as WordNet to identify synonyms and calculate similarity scores using cosine similarity. They retrieved relevant answers from a dataset collected from three books: Information and Computer Technology, Computer and Communication Technology, and Introduction to Computers. The system performance was evaluated using the F1 score, which achieved 87%.

The authors of [12] (2023) introduced a dual COBERT algorithm, consisting of a retriever and a reader. Its purpose was to answer complex questions by searching a database of 59,000 documents related to the Coronavirus, provided by the CORD-19 initiative. The retriever utilizes the TF-IDF vectorizer to select the top documents based on their scores, while the reader employs a BERT model pre-trained on the SQuAD 1.1 development dataset. The results achieved an exact match score of 80% and an F1 score of 87%.

The authors of [13] (2023) presented a Question Answering System that employs vectorization techniques, TF-IDF, and statistical scoring methods (cosine similarity approach) for document retrieval, allowing it to provide accurate answers to natural language questions. The researchers utilized the BNP Paribas dataset and evaluated the system's performance based on the F1 score, achieving an F1 score of 88% and an exact match of 80%.
Finally, the authors of [14] (2024) presented JMLR (Jointly trained LLM and Information Retrieval), a novel approach that integrates a large language model (LLM) with an information retrieval (IR) model during the training phase. This synchronized training mechanism enhances JMLR's ability to retrieve clinical guidelines and utilize medical knowledge for reasoning and answering questions, while also reducing the demand for computational resources. JMLR was evaluated based on the percentage of correct answers, achieving a score of 70.5%.

After reviewing the related works discussed above, the objectives can be summarized into two main categories. Some studies focus on generating questions and their corresponding answers from a given passage [10]. Although that study involved the question generation task, they did not address the validity of the generated questions in terms of grammar, semantics, and comprehension. Other studies aimed to retrieve answers, which were typically short and merely represented an information retrieval task [11] [12] [13] [14]. In contrast, this research is distinct from traditional information retrieval models. It seeks to tackle the challenge of generating valid and intelligible answers as a complete sentence in a human language. To achieve this, it performs two consecutive tasks: first, it retrieves relevant information, and then it expands upon that information to create a coherent sentence that is understandable to humans.

This research develops a new algorithm for "Answers Generation based on English Textual Analyzer (AGETA)," which aims to fill the gaps in answer generation by applying an English text analyzer to the input texts that have been preprocessed to extract relevant information. Then, a deep learning model (T5) is used to generate short answers, then expanded to long, perfect answers by applying the grammar mechanism built based on the concept of (Context-Free-Grammar). Finally, it ensures the answers generated are clear and coherent and produce more reliable results in practical domains.

## 2. Dataset

Passages and related questions were chosen from the Stanford Question Answering Dataset (SQuAD 2.0), which contains 107,785 question-answer pairs based on 536 passages derived from Wikipedia articles [15] [16]. The SQuAD 2.0 dataset is selected because it features a diverse range of Wh-questions (such as what, where, when, who, which, and how) that address various aspects of information. These questions certainly reflect a human-like style of analysis, and require the generation of long, coherent, and grammatically correct answers, making them ideal for assessing AGETA. The ability to produce complete and grammatically correct sentences is essential to the success of AGETA, and the system performs a thorough test of all its components based on these criteria.

As of February 2025, the online database cannot provide comprehensive and detailed answers based on the surveys conducted by researchers. Due to this limitation, researchers required the assistance of a human expert to deliver complete and grammatically coherent responses. However, because of the limited time, researchers could only choose a small number of carefully selected sentences. The researchers carefully selected passages covering a variety of topics, along with relevant questions. This focused selection does not diminish the challenges that AGETA encounters; it remains a demanding assessment of its capacity to generate detailed and linguistically sophisticated answers. As a result, 50 different Wh-questions were chosen from 14 dataset sections to highlight all the important aspects of AGETA. Initial experiments indicated that increasing the number of questions did not impact the accuracy of the results, making it unnecessary to broaden the tester . Figure 1 shows an example of a passage with its set of question-answer. There are five questions about information implied within the passage. Label Q refers to a question. All questions are answered with the short answer labeled A.

In 1969, Neil Armstrong became the first person to wa
the moon during NASA's Apollo 11 mission. The mission w
unched from Kennedy Space Center in Florida, and the histo
nding occurred on July 20th. Armstrong's famous words, "Tha
e small step for man, one giant leap for mankind," w
oadcast to millions worldwide. The astronauts traveled to t
oon using the Saturn V rocket, and after completing th
ssion, they safely returned to Earth, landing in the Paci
ean.

: What did Neil Armstrong do?
: walk on the moon
: first person to walk on the moon

: Where did the mission launch from?
: Kennedy Space Center in Florida
: was launched from Kennedy Space Center in Florida

: When did the moon landing occur?
: July 20th
: The landing occurred on July 20th.

: How did the astronauts travel to the moon?
: Using the Saturn V rocket
: to the moon using the Saturn V rocket

: Who was the first person to walk on the moon?

**Figure 1**: an example of a passage with its set of question-answers.

## 3. Preliminaries

This section explains the methodologies employed in this research:

- *Spacy and NLTK*: known for their efficiency, easy to use, and powerful tools in NLP, such as encryption, named entity recognition, and part-of-speech tagging. It accurately identifies part-of-speech categories, resulting in a document with entities and their corresponding POS tags. This research was utilized during the textual analysis phase [17] , [18].
- *T5 model*: (Text-to-Text Transformer) a pre-trained model on various types of data that consists of an encoder and decoder models. It gained popularity due to its outstanding performance in natural language processing, ease, competitive performance, and solving tasks such as text-to-text mapping problems [19]. It uses bidirectional context to understand the input text and autoregressive decoding for the output text generation. Therefore, it is utilized to extract proper and contextually correct short answers from the retrieved documents [20]. In this research, T5 is used in the 4$^{th}$ phase to generate short answers.
- *TFIDF Vectorizer*: (Term Frequency-Inverse Document Frequency) is a term-weighting schema widely used in recommender systems, search engines, and information retrieval. It ranks documents by evaluating the significance of a word in a document relative to a set of documents [21]. The TF-IDF procedure combines two statistics: term frequency and inverse document frequency [22]

$$W_{t,\mathrm{d}} = tf_{\mathrm{t,d}} \,.\log\left(\frac{N}{df_t}\right) \tag{1}$$

Term frequency measures how often a term (t) appears in a specific document (d). In contrast, inverse document frequency assesses how much information a term provides to a document. It is calculated by taking the logarithm of the total number of documents in a corpus (N) divided by the number of documents that contain the term (t). In this research, *TFIDF* is used in the 3$^{rd}$ phase as the first step for information retrieval.

- *Cosine similarity*: a mathematical metric used to measure the similarity between two non-zero vectors. It is widely applied across various domains, such as neural networks, text classification, medical diagnosis, and big data processing. It is particularly valued for its ability to provide a bounded similarity measure independent of vector magnitude, making it robust for diverse applications [23]. Cosine similarity can be defined as the equation [24] :

$$cosine(V,\mathrm{W}) = \frac{V \,.\, W}{|V||W|} = \frac{\sum_{i=1}^{N} V_i \; W_i}{\sqrt{\sum_{i=1}^{N} V_i^2}\;\sqrt{\sum_{i=1}^{N} W_i^2}} \tag{2}$$

V and W are two vectors, each with a length of N. The cosine value of the angle between these vectors ranges from 1 (when they point in the same direction) to 0 (when they are orthogonal) and down to -1 (when they point in opposite directions). However, because the raw frequency values are non-negative, the cosine values for these vectors range only from 0 to 1.

## 4. The Methodology of The Proposed System

The proposed method, AGETA, stands for "Answers Generation based on English Textual Analyzer." It attempts to generate perfect *long* answers by applying a sequence of phases (pipeline). Its phases included applying linguistic preprocessing, performing textual analysis, extracting the most_ similar_ sentence, and generating answers, as shown in **Figure 2**. Each phase involves a sequence of operations, and the internal results are the input to the next. The original input is helpful for the first phase as well as the fourth phase. Finally, the generated answers are ensured to be clear and coherent.

Before any of these four phases could be performed, the necessary task was human reviewing the provided dataset to verify the short answer to each question. Some **wrong answers** occurred at least two times, and was corrected.

### 4.1 Linguistic Preprocessing

This phase is applied after reading the texts, where all punctuation marks are removed except for periods at the end of sentences, the separator of the possessive pronouns " ' " and "," separator. Extra spaces and foreign characters were systematically removed, and all characters present within the texts **were** transformed into lowercase.

### 4.2 Textual Analysis

Before explaining this phase, it is important to highlight a significant and frequently employed technique in NLP, tokenization, which serves as a foundational procedure for numerous subsequent processes. Sentence tokenization and word tokenization are two types of tokenization. The first type splits the text into individual sentences, and the second splits the text into words. At this phase, a different series of NLP techniques are performed in both the passage and question individually using the Spacy and NLTK libraries.

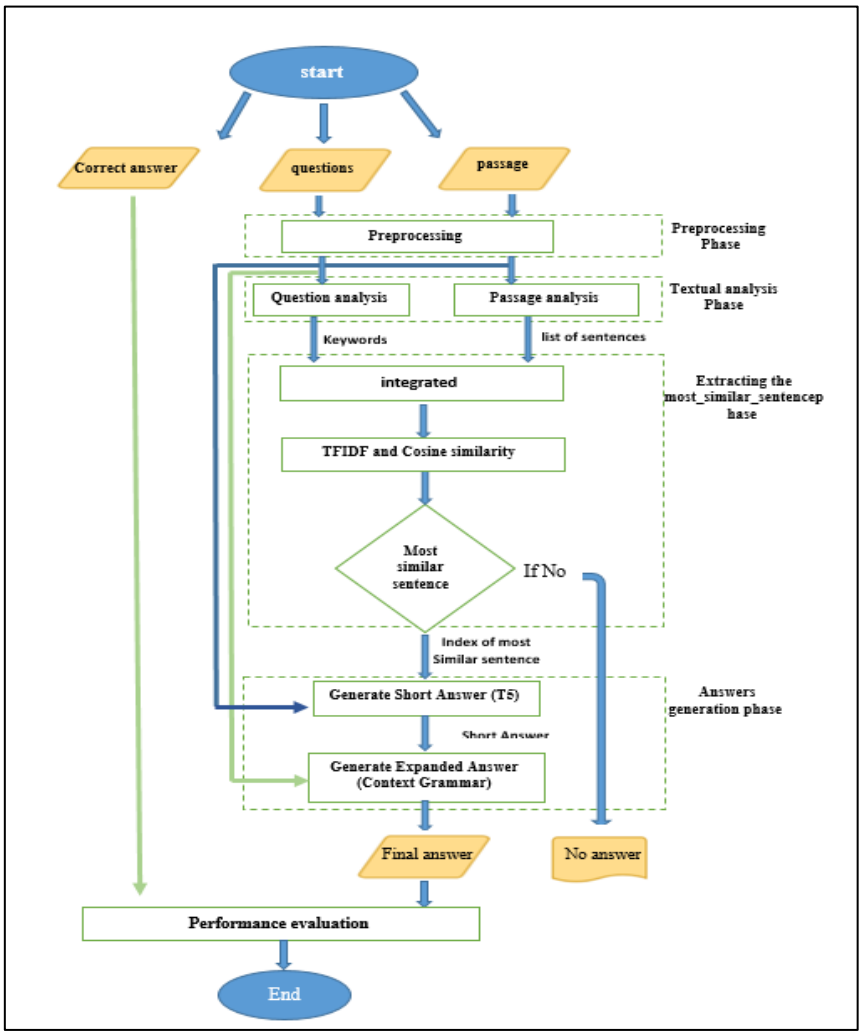


**Figure 2**: answer generation

### 4.2.1 Question Analysis

Question analysis was performed to extract keywords using a part-of-speech tagging (POS) technique. The initial step is tokenizing the question into words. Each token is classified into its part of speech and labeled through POS tags. Table 1 shows the most common POS tags available in the Spacy and NLTK libraries, and their descriptions and examples are added in the table for clarification. Specific types of tags were chosen based on the question types, as shown in **Table 2**. Ultimately, these entities are integrated to form a single sentence of keywords.

**Table 1***:* POS tags with Description and Examples

| *POS Tag* | *Description* | *Example* |
|---|---|---|
| NOUN | Noun | cat, car |
| PROPS | Proper Noun | John, London |
| PRON | Pronoun | he, they, it |
| VERB | Verb | run, eat, be |
| ADJ | Adjective | happy, tall, fast |
| ADV | Adverb | quickly, very |
| ADP | Preposition | in, on, at, over |
| DET | Determiner | the, a, an, this |
| CONJ | Coordinating Conjunction | and, but, or |
| SCONJ | Subordinating Conjunction | because, although |
| INTJ | Interjection | wow, oh, ouch |
| NUM | Number | one, two, 100 |
| PART | Particle | not, to |
| AUX | Auxiliary Verb | is, have, will |
| SYM | Symbol | $, %, @ |
| PUNCT | Punctuation | ., !, ? |

**Table 2***:* Wh-questions with entities

| *Wh-question* | *POS Tag* |
|---|---|
| What | NOUN, PROPS, VERB |
| Who | NOUN |
| Where | NOUN, PROPS, VERB |
| When | NOUN, PROPS, VERB |
| How | NOUN, PROPS, VERB, ADJ, ADV |
| Which | NOUN, PROPS, ADJ |
| Whose | NOUN, PROPS |

### 4.2.2 Passage Analysis

Techniques involve analyzing a passage and preparing it as a list of sentences for use in the next phase. Firstly, the passage is tokenized into words to remove stopwords within it. English stopwords within the NLTK library have been used to remove all stopwords. Then, the remaining word tokens should be integrated and returned to the form of a passage. Finally, the passage is tokenized into a list of sentences depending on the period at the end of each sentence.

Applying those techniques results in the keywords and list-of-sentences. Keywords are extracted from the input question, while list-of-sentences is extracted from the input passage. Then, those internal results must be integrated into one matrix to put it in a form appropriate to TFIDF that will be applied in the next phase.

### 4.3 Extracting the most_similar_sentence

This phase holds greater significance due to its capacity to yield a sentence that bears the most_ similar_ sentence to the sentence of keywords, compared to the array of sentences in the matrix. TFIDF victories convert the matrix into a scalar vector based on words and their frequency. Then, the Cosine similarity method is used to compute the similarity values and return the index of the most_similar_sentence. Similarity values are ranked in descending order; if all values are equal to zeros and no sentence is identical to the keywords, then (AGETA) exits and completes the remaining phase of the answer generation.

### 4.4 Answer Generation

The last phase of (AGETA) is the answer generation. In this phase, the Short Answer is generated first, and then the longer (expanded) answer is generated to give a clearer meaning to the answer sentence. (AGETA) is committed to making the sentences complete in terms of grammar and adhering to the verb tense in the question to make the answer have a clear meaning.

### 4.4.1 Generate Short Answer

The original input passage was tokenized into a list of sentences, and then the sentence with the same index as the most_ similar_ sentence was extracted to be the first input in this phase. The second input is the original question. The T5 model encodes these inputs and then decodes them to generate the Short Answer used in the next phase to generate the expanded answer.

### 4.4.2 Expanded Answer Generation

A preprocessed question and a short answer (Short A) are two inputs in this phase. At first, it tokenizes the question into words. Then, the following parts are extracted from the question:
•question tool
•question about (Ques_ab) one or more words that come after the question word to the auxiliary verb (if any)
•auxiliary verb (Aux_v) (if any)
•the rest of the question (Rest of q), which contains the Main verb (in some cases, the Main verb is isolated from the rest)

Then, the answer sentence is formed according to the following rules:
1-In the presence of an auxiliary verb (is, are, was, were) and others. **Table 3** shows the grammar of the special rules for each question tool.

**Table 3:** Grammars for the first rule

| *Question Tool* | *Expanded Answers* |
|---|---|
| What, Who, Which | (Short A) + (Aux_v) + (Rest of q) |
| Where, When | (Rest of q) + (Aux_v) + (Main v) + (Short A) |
| How | (Ques_ab) + (Aux_v) + (Main v) + (Short A) + (Rest of q) |
| How many, How much | (Short A) + (Ques_ab) + (Aux_v) + (Rest of q ) |

2- In the presence of an auxiliary verb (do, did, does):
In this case, auxiliary verbs are omitted from the answer sentence; the Main verb in the sentence is processed based on the type of auxiliary verb. If the auxiliary verb is “did,” it is changed to the past tense. If it is “does,” the third person "s" is added to the verb. If it is “do,” the verb form is not changed. Table 4 shows the grammar of the special rules for each question tool.

**Table 4:** Grammars for the second rule

| *Question Tool* | *Expanded Answers* |
|---|---|
| What, Which, Who | (Rest of q) + (Short A) |
| Where, When | (Rest of q) + (Prepos) + (Short A) |
| How | (Rest of q) + (Prepos) + (Short A) |
| How many, How much | (Rest of q) + (Short A) + (Ques_ab) |

3- In the absence of any auxiliary verb, **Table 5** shows the grammar of the special rules for each question tool.

**Table 5:** Grammars for the third rule

| *Question Tool* | *Expanded Answers* |
|---|---|
| What, Which, Who | (Short A)+ (Ques_ab) |
| Where, When | (Ques_ab) + (Prepos) + (Short A) |
| How | (Ques_ab) + ( Short A) |
| How many, How much | (Short A) + (Ques_ab) |

Note: (Prepos) is denoting preposition, adding "in" if there is not one with the sentence of where, "on" with the sentence of when, and "by" with the sentence of How.
Finally, the expanded answer is ensured to be formatted well, no repeated adjacent words result from merging the short answer with the answer sentence, and it ends appropriately with a period to help maintain a consistent and polished presentation of the final output.

## 5. Evaluation

Two different types of accuracy measurement for AGETA were used. In the text generator domain, it is fairer to evaluate accuracy depending on the consistency, overall meaning, completeness, etc. Therefore, the first type used was human performance using Equation 3 for short answers and Equation 4 for long answers. The second type used Sentence Transformers (BERT-based models). It is a model that relies on contextual embeddings to understand semantic relationships, and it is used to compare long sentences and find similarity values between sentences [24], [25].

## 6. Results and discussion

AGETA attempts to generate a correct answer as a complete comprehension sentence. The previous works at [12], and [13]mentioned in the introduction represent a similar approach; they used the same dataset. The methods used to extract relevant information and generate answers were similar, as they relied on TF-IDF, cosine similarity, and deep learning techniques to generate answers. However, none of these studies succeeded in expanding the short answer into a long answer that was meaningful and coherent in terms of logic, as they relied on specific rules they developed to expand answers, without achieving a sufficient level of understanding and linguistic fluency.

AGETA receives a passage with its list of questions as English unstructured typed texts. It is a big challenge to analyze unstructured texts. The behavior of AGETA was followed by applying it to the chosen questions mentioned in section 3. There are a variety of (50) English Wh-questions related to (14) passages about different topics [26].

*An example* of one passage with its question is presented. **Figure 3** shows an example of a "Doctor Who" passage with one question. To study AGETA's behavior towards understanding the input text and its ability to generate consistent sentences, internal results of the example will be shown by figures, step-by-step.

- Phases (1 and 2): linguistic preprocessing and textual analysis are performed on the texts. Figure 4 shows the results.
- Phase 3: extract the most_similar_sentences and their index as shown in Figure 5.
- Phase 4: generate short answers and finally generate the expanded answer as shown in Figure 6.

Twelve actors have headlined the series as the Doctor. The transition from one actor to another is written into the plot of the show, as well as the differing approach to the role that each brings, under the concept of regeneration into a new incarnation. The show's premise is that this is a life process of Time Lords through which the character of the Doctor takes on a new body and, to some extent, new personality, which occurs after sustaining an injury which would be fatal to most other species. Each actor's portrayal differs, but they are all intended to be aspects of the same character and form part of the same storyline. The time-travelling nature of the plot means that, on occasion, different Doctors have met each other. Peter Capaldi took on the role after Matt Smith's exit in the 2013 Christmas special "The Time of the Doctor".

Q: Which episode was Matt Smith's final one as Doctor Who?

**Figure 3**: the passage and question

**Question**
the keywords are: ['episode', 'matt', 'smith', 'one', 'doctor']
**passage**
Sentence 0: twelve actors headlined series doctor
Sentence 1: transition one actor another written plot show, well differing approach role brings, concept regeneration new incarnation
Sentence 2: show 's premise life process time lords character doctor takes new body, extent, new personality, occurs sustaining injury would fatal species
Sentence 3: actor 's portrayal differs, intended aspects character form part storyline
Sentence 4: timetravelling nature plot means, occasion, different doctors met
Sentence 5: peter capaldi took role matt smith 's exit 2013 christmas special time doctor

**Figure 4:** keywords and list of sentences

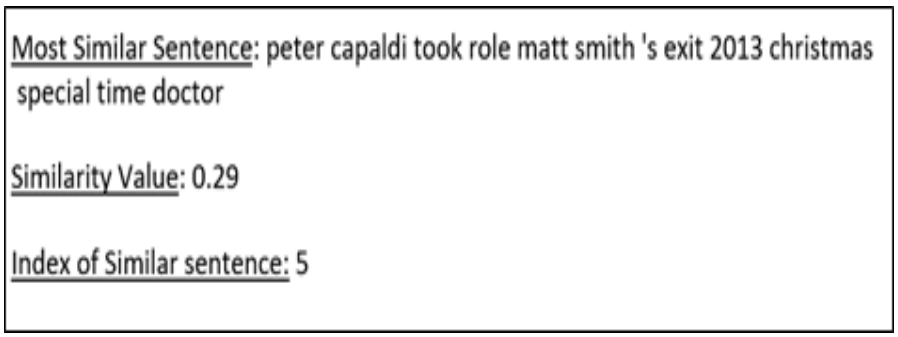


**Figure 5**: most_similar_sentence, similarity value,and the index of a similar sentence

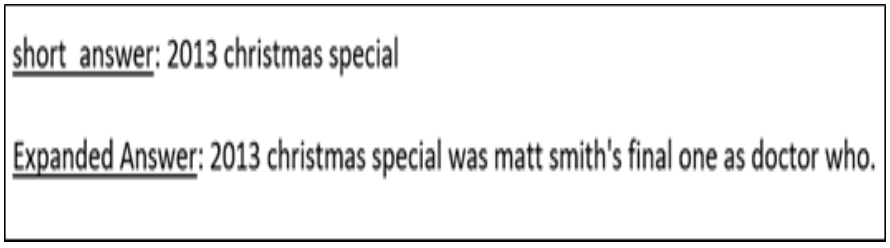


**Figure 6**: generated answers

AGETA's perfect behavior toward understanding and generating English sentences is based partially upon its perfect preprocessing phase. It was flexible to keep punctuation that was important for two tasks: understanding English text and facilitating processes of the following phases. There were many experiments before keeping some punctuation and after keeping them. Another reason for the perfect behavior was keeping the original version of the texts (the question as well as the passage) without discarding them and using them where they were needed. Encompassing the original version of the text with the internal results at the answer generation phase added a strong point and improved results.

The results for the fifty tested questions (shown in Appendix A) [26] as two parts: the first part is the short answer generator's results, and the second part is the expanded answer generator's results. As the first part explains, among the fifty short answers generated, there are four wrong answers (9, 14, 35, and 47 in Appendix A). According to the human accuracy calculation outlined in Equation 3 (Exact Matching), which was utilized in both research [12]and [13], AGETA demonstrated superior performance in generating short answers, achieving an accuracy rate of 92%. In comparison, the success rate for both related works was 80%, as shown in Table 6. The correct answers included different types of questions (who, how many, what, where, which, when), while wrong answers included (what, where, when). Duplicating question-types of wrong answers at question-types-list of correct answers proves that AGETA has no limitations against a specific type of question. All incorrect answers are marked as no direct typed answer in the passage; instead, their answers could be predicted from understanding the overall meaning of the passage. Otherwise, the answer may require background information about the passage's subject.

$$\text{Exact match} = \left(\frac{\text{RA}}{\text{NQ}}\right) * 100 \tag{3}$$

RA is the Right Answer, and N Q is the number of questions.

**Table 6:** The human accuracy between two types of research and AGETA

| Research | Work | Dataset | Result |
|---|---|---|---|
| ***Author in [12]*** | *COBERT algorithm* | *SQuAD* | *Exact match=80%* |
| ***Author in [13]*** | *Question answering system* | *BNP Paribas* | *Exact match=80%* |
| ***This research*** | *AGETA* | *SQuAD* | *Exact match=92%* |

For more details about the reason for wrong answers, the implementation of AGETA was tracked step by step. It was ensured that the keyword extractor worked correctly. All internal results from the second phase (question-keywords and passage-list-of-sentences) were integrated correctly into one matrix, which was appropriate for TFIDF. The wrong way at the most_similar_sentence extractor is the cosine similarity method. In other words,, the limitation was because the cosine similarity method failed with comprehension questions that depended on combining information from several sentences or on common sense information not explicitly included in the passage. This seems fair if we remember that the Cosine similarity method treats the text as a matrix of numbers. So, it succeeds in the case of information that is directly available but fails in the case of information that must be inferred from the content.

For more clarification, let's study question (47 in Appendix A). It is clear that the most prior keywords are “moon landing”. Figure 7 shows the set of sentences related to the question. Sentence (b) includes the correct answer. But Cosine similarity could not understand that because the keyword “moon” is not found here, and it did not find any reference to the “moon”. Really, sentence (a) joined “landing” with “moon,” but this comprehension is beyond cosine similarity abilities. So, it is believed that sentence (c) is what they wanted. Because of finding the keywords “moon” and “landing”.

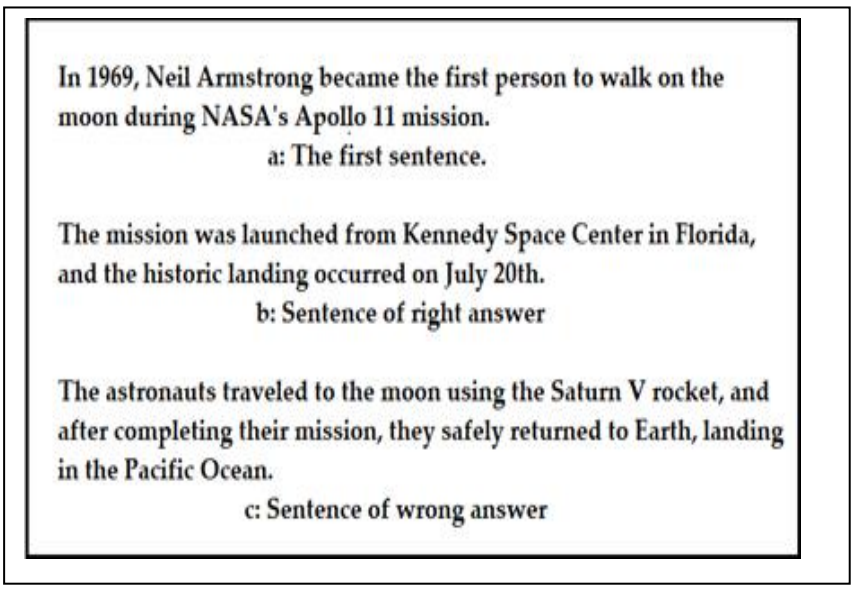


**Figure 7**: Clarifying wrong short

The AGETA short answer generator demonstrates an accuracy rate of 92%, which means the error rate is 8%. It's important to note that this is not the final phase of the AGETA process but rather a preliminary step that leads to the final phase.

As the second part explains, it is fair to test the results of the (46) right short answers isolated from the others. AGETA succeeded in expanding short answers into the proper consistent English sentences for (43). AGETA expanded short answers into English sentences, with verb-tense wrong for (3) (10, 22, and 46 in Appendix A). It constructed sentences by applying English rules of question-answer shown in Tables 3, 4, and 5. So AGETA benefits from words in the original question. AGETA succeeded for all, although there are many questions encompassed a foreign remain-of-question with the keyword as an example (5 at appendix A). In other words, some questions asked about something real in the passage but were followed by terms taken from another sentence in the passage. AGETA benefited from the Short Answer and expanded it, keeping remain-of-question. Comparing AGETA's expanded answers to the human expert's expanded answers, it is found that sometimes the human answer differs in that he expanded using words from the wanted sentence in the passage instead of keeping the remain-of-question as done by AGETA.

So, in many questions, AGETA's long answers have the same overall meaning as the human answers, such as (7, 11, 12, 28 in Appendix A) with simple word differences. For more clarification about the wrong verb that occurred in (10, 22, and 46). AGETA failed to concatenate the “third person singular s” with the Main verb “encompass” at (10). AGETA was unable to use the past tense with the Main verbs “make” and “launch” at (22 and 46). However, AGETA succeeded in formulating the right verb tense at other expanded answers, for example, (45, 48, and 50).

From all the above, one can see that the AGETA expanded answer generator works with accuracy reaching 96.7% according to Equation 4 when isolating the wrong short answer questions, and there is no mentioned semantic error. For the 4 wrong short answers, AGETA succeeded in generating expanded wrong answers that were grammatically correct. It generated the proper English sentence for the wrong answer. Because of the aims of AGETA to generate English sentences adding to generating expanded answers, a correct answer with right English syntax is given 1, while a correct answer with wrong English syntax is given 0.5, and then a wrong answer with right English syntax is given 0.5, and there are no solutions with wrong answer and wrong English syntax.

$$\text{accuracy} = \left( \left( \frac{\text{R A}}{\text{N Q}} \right) + \left( \frac{\text{H R}}{\text{N Q}} \right) * 0.5 \right) * 100 \qquad (4)$$

RA is the Right Answer, H R is Half Right, and N Q is the number of questions.
All the above was the human evaluation task. The following is the evaluation of AGETA using Sentence Transformers (BERT-based model). Accuracy was applied for each question answer alone, the similarity was measured between the human expert answer and the AGETA answer, and then the average for each passage was calculated. It ranged from (81%) to (97%), and the overall average was (90%). Although the transformer model is classified as a similarity measurement that relies on contextual embedding to understand semantic relationships, the similarities were unbelievable for many passages; for example, the “Apollo program” passage accuracy was 93%, while its three AGETA answers were almost identical to the human expert answers.

## 7. Conclusion

Generating answers with complete comprehension and consistent sentences is a big challenge, especially when the given information is unstructured text (passage). Manipulating the input text via the known preprocessing techniques, applying POS tagging, extracting

keywords, extracting the most_similar_sentence, generating short answers, and then generating expanded answers, are developing a perfect approach AGETA for answer generation. From all the experiments implemented by AGETA, many conclusions can be drawn.

Firstly, and before any preprocessing, AGETA advised verifying the original dataset. It may include some errors that cause bad accuracy of any proposal. Then, be careful while applying the traditional preprocessing techniques; sometimes, punctuation marks are considered an important element, especially in the domain of text generation.
AGETA advises keeping the period at the end of the sentence, the possessive s, and the comma used for listing elements of a set. Adding to all that, be careful not to damage the original input text; it may be useful at the next step.

AGETA advises applying POS tagging, which facilitates the analyses the generation processes. Also, it achieved great results by extracting keywords and applying the cosine similarity method. On the other hand, AGETA showed that the Cosine similarity method may be weak with questions that require an understanding of overall meaning and background information. AGETA chose the deep learning model (T5) successfully to perform the short answer generation task, which effectively passed all the questions that passed the previous steps.

AGETA has introduced innovative methods for transforming short answers into coherent long answers, addressing a challenge not fully resolved in previous studies. By structuring grammar mechanisms successfully, they coherently organized the English sentence rules and did not accept any errors.

Anyone may think Why a long answer? it is clear that when a person wants to get an answer, it is sufficient to get a relevant piece of information, which is the short answer; whereas when the greater ambition —to serve as a starting point for an automated conversational model—a valid intelligible sentence will be a critical requirement. Therefore, AGETA is here, and future work must be directed towards enhancing it to serve that ambition. The first one was to improve the processing tools used at the step marked as the error stage, which was the retrieval step. Deep learning models, particularly transformer-based models, could be used for sentence retrieval. Additionally, incorporating neural syntax checking may improve long-answer generation, while investigating techniques to address complex, multi-step reasoning questions would be beneficial.

**Acknowledgments**
We are grateful to the English language expert for his assistance in this research. Also, extended to the College of Science and the Presidency of Al-Nahrain University for their approval of this research.
**The authors, in the end, declare that they have no conflicts of interest.**

**References**

**[1]** R. Barskar, G. F. Ahmed, and N. Barskar, "An Approach for extracting exact answers to question answering (QA) system for English sentence," *Procedia Engineering,* vol. 30, pp. 1187-1194, 2012.

**[2]** Z. T. Ke, P. Ji, J. Jin, and W. Li, "Recent Advance in Text Analysis," *Annual review of statistics and its application,* pp. 372-347, 2023.

**[3]** Mark Prrkins, "Approaches to Text Analysis," *Global Language Review,* vol. 4, no. 1, pp. 1-7,

2019.

**[4]** Abeer K. Al-Mashhsdany, Abdulwadood K. l-Mashhadany, Waleed K. Al-Mashhadany, "Root-Steam Approach in General Analyzer System for Arabic Language ( RSGAS)," *J. of University of Al Anbar for pure science,* vol. 10, no. 3, 2016.

**[5]** Abeer. K. Al-Mashhsdany, S. A. Ahmed, "Textual Analysis Application: Subject Review," *J. of University of Al Anbar for pure science,* vol. 12, no. 3, pp. 71-83, 2018.

**[6]** Kuldeep Vayadande, Preeti A. Bailke, Lokesh Sheshrao Khedekar, Rakesh Kumar, Varsha R. Dange, A Review on Text Analysis Using NLP, USA: Wiley, 2024.

**[7]** Wenhao Yu, Chenguang Zhu, Zaitang Li, Zhiting Hu, Qingyun Wang, Heng Ji, & Meng Jiang, "A survey of knowledge-enhanced text generation," *ACM Computing Surveys,* vol. 54, no. 11, pp. 1-38, 2022.

**[8]** Vincent Claveau, "Neural text generation for query expansion in information retrieval," in *ACM International Conference on Web Intelligence and Intelligent Agent Technology*, 2022.

**[9]** P. Upadhyay, R. Agarwal, S. Dhiman, A. Sarkar, and S. Chaturvedi, "A comprehensive survey on answer generation methods using NLP," *Natural Language Processing Journal,* vol. 8, p. 100088, 2024.

**[10]** M. Roemmele, D. Sidhpura, S. DeNeefe, Ling Tsou, "Answer Quest: A System for Generating Question-Answer Items from Multi-Paragraph," in *Association for Computational Linguistics*, 2021.

**[11]** Almotairi, M., & Fkih, F, " Developing a Semantic Question Answering System for E-Learning Environments Using Linguistic Resources," *Journal of Education and E-Learning Research,* vol. 9, no. 4, pp. 224-232, 2022.

**[12]** Alzubi, J.A., Jain, R., Singh, A., "COBERT: COVID-19 Question Answering System Using BERT," vol. 48, p. 11003–11013, 2023.

**[13]** Manjunath, T. N., Yogish, D., Mahalakshmi, S., & Yogish, H. K, "Smart question answering system using vectorization approach and statistical scoring method," in *Materials Today: Proceedings*, 2023.

**[14]** A. Majumdar, A. Ajay, X. Zhang, P. Putta, S. Yenamandra, M. Henaff, "OpenEQA: Embodied Question Answering in the Era of Foundation Models," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

**[15]** Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang, "SQuAD: 100,000+ Questions for Machine Comprehension of Text," in *Conference on Empirical Methods in Natural Language Processing*, Austin, Texas, 2016.

**[16]** Pranav Rajpurkar, Robin Jia, and Percy Liang, "Know What You Don't Know: Unanswerable Questions for SQuAD," in *56th Annual Meeting of the Association for Computational Linguistics*, Melbourne, 2018.

**[17]** Mark Neumann, Daniel King, Iz Beltagy, and Waleed Ammar, "ScispaCy: Fast and Robust Models for Biomedical Natural Language Processing.," in *Proceedings of the 18th BioNLP Workshop and Shared Task*, Italy, 2019.

**[18]** M. Wang and F. Hu, "The Application of NLTK Library for Python Natural Language Processing in Corpus," *Theory and Practice in Language Studies,* vol. 11, no. 9, pp. 1041-1049, 2021.

**[19]** Jianmo Ni, Gustavo Hernandez Abrego, Noah Constant, Ji Ma, Keith Hall, Daniel Cer, and Yinfei Yang, "Sentence-T5: Scalable Sentence Encoders from Pre-trained Text-to-Text Models.," in *Association for Computational Linguistics*, 2022.

**[20]** Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., ... & Liu, P. J., "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of machine learning research,* vol. 21, no. 140, pp. 1-67, 2020.

**[21]** Joeran Beel, Stefan Langer, Bela Gipp, "TFIDF: A Novel Term-Weighting Scheme for User Modeling based on Users 'Personal Document Collections," in *Proceedings of the conference*, Wuhan, China, 2017.

**[22]** Grootendorst, M., "BERTopic: Neural topic modeling with a class-based TF-IDF procedure,"

*arXiv preprint arXiv:2203.05794.,* 2022.

**[23]** Putri Yuni Ristanti, Aji Prasetya Wibawa, Utomo Pujianto, "Cosine Similarity for Title and Abstract of Economic Journal Classification," in *5th International Conference on Science in Information Technology*, Yogyakarta, Indonesia, 2019.

**[24]** Lukas, Stankevičius., Mantas, Lukoševičius, "Extracting Sentence Embeddings from Pretrained Transformer Models," *Applied Sciences,* vol. 14, no. 19, pp. 8887-8887, 2024.

**[25]** R. Kora., A. Mohammed, "A Comprehensive Review on Transformers Models for Text Classification," in *2023 International Mobile, Intelligent, and Ubiquitous Computing Conference*, Cairo - Egypt, 2023.

**[26]** Appendix A, [Online]. Available: https://drive.google.com/file/d/1SSR-KbD9luF-esm6cIj4bePVKjU_ly6C/view?usp=drive_link.