Contextual Query Expansion: Transforming Question Answering with an MLM-Based Approach Using A Transformer Model

Muhammad Manzoor Faisal
Department of Computer Science, COMSATS University Islamabad, Sahiwal Campus, Sahiwal, Pakistan
manzorch77@gmail.com

Javed Ferzund
Department of Computer Science, COMSATS University Islamabad, Sahiwal Campus, Sahiwal, Pakistan
jferzund@cuisahiwal.edu.pk

Ahmad Shaf
Department of Computer Science, COMSATS University Islamabad, Sahiwal Campus, Sahiwal, Pakistan
ahmadshaf@cuisahiwal.edu.pk (corresponding author)

Afnan Aldhahri
Department of Software Engineering, College of Computing, Umm Al Qura University, Makkah, Saudi Arabia
amdhari@uqu.edu.sa

Received: 4 October 2025 | Revised: 12 November 2025 and 5 December 2025 | Accepted: 7 December 2025

 DOI: https://doi.org/10.48084/etasr.15306

ABSTRACT

The purpose of Question Answering (QA) frameworks is to provide precise answers to Natural Language Queries (NLQ) by extracting relevant information from large document collections. A key challenge is that users often struggle to formulate optimal queries, a limitation that prior methods have often failed to address by neglecting the valuable role of contextual information. To overcome this, this study introduces an MLM-based approach that takes advantage of contextual cues from document collection. By harnessing the context extraction capabilities of transformer models (such as BERT), expanded queries are automatically generated to better capture user intent. The proposed technique significantly improves the effectiveness of QA. The proposed MLM-generated expanded queries outperform query expansion methods based on WordNet, validating the integration of context clues as a promising development to refine the accuracy and relevance of QA responses.

Keywords-questions; answers; query expansion; natural language processing; information retrieval; MLM; BERT; contextualize query expansion

I. INTRODUCTION

It is an important and challenging task to provide a useful and efficient way for end-users to obtain information. Keyword-based search engines have achieved immense popularity by supporting easy access to relevant websites across thousands of files. However, when only a few terms are used to define the purpose of the question, keyword search cannot be descriptive enough, and the returned results are only a collection of indexed documents containing input words instead of a direct response [1]. Information Retrieval (IR) for Question Answering (QA) is an approach that enables end-users to communicate in natural language and provide the correct pieces of information to receive answers [2]. QA systems are described by typical high-level architectures. First, a natural language query is handled, and answers are derived to explain what has been questioned [3, 4]. Then, a knowledge base or record set is queried using these suggestions to obtain specific information [5, 6]. The information is then analyzed to choose the most accurate solution, concisely articulated, among the more feasible alternatives [7, 8]. These systems usually address questions divided into two typologies (factoid or non-factoid). The identification of a factoid question is answered by specific keywords expressing a contextual term or a place, while phrases or paragraphs expressing meanings, explanations, or methods can address non-factoid questions.

QA structures can be further categorized into two key methodologies, including IR-based [9, 10] and knowledge-based [11], varying in the categorization of information sources and, therefore, on whether IR can be carried out on them. Furthermore, they may be distinguished between open-domain, based on the typology of questions to be asked, if no constraint is created on the query field, and closed-domain, if queries are restricted to a particular area.

In [12], the focus was on IR-based QA structures on closed-domain questions, answering massive datasets, and explicitly extracting responses to factual queries, motivated by the fact that existing knowledge repositories are far from complete and comprehensive. Evidence does not always exist in a standardized manner to answer the questions. In general terms, the output of QA systems is specifically related to their ability to recover appropriate sentence collections that balance natural language issues. Usually, this activity is carried out by extracting lexical features from each query and deciding the probable Lexical Answer Type (LAT) and related keywords. This data is compiled into an IR engine query that makes it feasible to use candidate sentences that are derived from document sets, consistent with the LAT and associated with keywords and entity names. When additional natural language queries utilize words from the domain in a format similar to that found in texts, many phrases retrieved are likely to fit well and include applicant replies.

However, due to synonymy, hypernym, and polysemy in QA and more broadly in IR activities, the answers differ in matching from a terminological point of view, causing the so-called word inconsistency issue. In the case of synonymy, individual definitions in both the questions and the documents may be interpreted by different terms. More common or unique terms appear in all queries and records in the event of a hypernym. The comparison of a particular term is found in all queries or records in the case of polysemy, but with several interpretations. This uncertainty, in addition to the fact that various individuals articulate their questions in varying methods, can lead to weak recovery performance [13].

Open-domain QA involves a broad exchange of data. This issue can be mitigated, as several documents significantly increase the chance of having at least one paragraph containing an applicant answering a question using the same words. Conversely, in closed-domain QA structures running on limited content, only a single sentence covering a candidate's response can be found, and the exact language could vary partially or entirely from that used in the question.

Query Expansion (QE) techniques have been suggested as a solution using document terms. Inconsistency issues in QA systems [14, 15] are resolved using various databases and automated methods [16]. In essence, both aimed at constructing IR queries by enriching words derived from questions in natural language. The externally relevant ones in terms of semantics, specifically synonyms or hypernyms, maximize the chance of matching the basic types of the words containing the responses to be obtained as they appear. Also, current QE methods are mostly based on the premise that, depending on their similarities, each query term will choose suitable leaders for its extension. Still, none of them takes the opposite view of the relation between these chosen expansion terms and the question's semantics. Recent efforts have been proposed to resolve the aspect of predicting terms, evaluating the validity of question training sets concerning specific models.

Previous methods lack the use of contextual meaning in the formation of QE. This study introduces a novel strategy for query augmentation using contextual information through an MLM-based model. BERT's word embeddings are used to locate relevant terms in the QE system, comparing its performance to that of WordNet. The BERT-based Query Expansion (QE) system outperforms the static WordNet embeddings. Since transformer models, such as BERT, are better at context extraction, the proposed approach uses them to generate expanded questions based on contextual information, offering better results in QA tasks. The objective of this study was to develop an approach that provides the desired result/answers to the user against his/her contextual clue or query information. The research questions were as follows.

- How to obtain relevant contextual information from large data.
- How to determine the exact keywords to find the actual desired data.
- How to use NLP and ML models to enhance common user search query results.

II. RELATED WORK

Word embedding approaches for QE have been extensively studied. In [17], Word2vec and GloVe were compared on the TREC newswire and ClueWeb datasets, demonstrating, through 10-fold cross-validation, that local embeddings provide stronger similarity measurements than global embeddings for QE. In [18], a model was developed to generate search queries from community questions using five L2R methods (ListMLE, LambdaRank, ListNet, RankBoost, and SVM-Rank). Sequential forward selection identified optimal feature sets, with results indicating that named entities were crucial for constructing effective two-to-five-term queries.

In [19], a QE technique used term embeddings to retrieve semantically related words based on vocabulary proximity. Tested on three CLEF corpora, this approach outperformed baseline methods, improving recall and precision without relevance feedback. The study in [20] combined NER and POS tagging for simultaneous QA and generation using 8 million Wikipedia sentences. This approach linked QG with QA to produce test pairs, showing promising results. The study in [21] used WordNet, domain-specific clustering, and semantic analysis with POS-based question pattern analysis. Evaluated on the 20-Newsgroup and TREC-9 datasets, this SWAG model outperformed several baselines. Neural-QA [22] is a large-scale QA library that segments long documents and expands queries using masked language models, providing a scalable infrastructure for developers and business applications.

In [23, 24], hybrid QE approaches combined word embeddings with lexical resources. For Italian QA systems, synonyms and hypernyms from MultiWordNet were extracted and contextualized using corpus relevance, improving the

accuracy of factoid answer extraction in the cultural heritage domain. An Arabic QA system [25] integrated question processing and document retrieval using SVM-based question classification (Li and Roth's taxonomy) and retrieved information from Arabic Wikipedia. Query expansion was combined with Arabic POS tagging and WordNet, outperforming existing techniques on the TREC and CLEF datasets. Another Arabic QA system [26] compared three classifiers (DT, SVM, NB) for question classification without translation, achieving 84% accuracy with SVM. Using Arabic Wikipedia and Google API for document retrieval, this approach outperformed translation-based methods. The study in [27] proposed a QE method to address the limitation that existing approaches use identical information sources and weighting for both single and compound queries, poorly capturing inter-term relationships. This method leveraged Wikipedia and WordNet as complementary information sources. In [28], a named entity disambiguation mechanism used an adjusted Lesk similarity measure between queries and Wikipedia disambiguation articles. This approach expanded short user queries containing ambiguous entities (names, places) for open-domain QA.

The effectiveness of QE has been explored across different approaches. In [29], QE was tested within the PLBR logical model using WordNet-derived linguistic information, finding that while lexical expansion generally did not improve retrieval, 30% of queries benefited when expansion terms were more general and widely-used than original terms while avoiding noisy phrases. A hybrid QE technique for Italian factoid QA [24] generated synonyms and hyponyms from lexical taxonomies, filtered and prioritized based on corpus relevance and semantic similarity to reduce noise from irrelevant terminology. SIRSD (Semantic Information Retrieval in Sports Domain) [30] reformulated queries using domain ontology and WordNet for context disambiguation, demonstrating higher precision and recall than traditional search methods, reducing semantic interoperability issues.

Several studies have explored NLP tasks for low-resource languages and LLM applications. In [31], a knowledge-based WSD approach for Hindi used the LESK algorithm with Hindi WordNet, achieving 71.4% accuracy. An ensemble MT approach for English-to-Hindi translation [32] demonstrated improved quality over single-engine baselines. In [33], NLP models were surveyed for converting natural language to SQL queries, enabling database access for non-expert users.

III. MATERIALS AND METHODS

A. Dataset Details

This study used the publicly available SQuAD v2 dataset [34]. SQuAD (Stanford Question Answering Dataset) is a dataset for question answering tasks, composed of more than 100,000 Questions and their answers. These questions and answers are extracted from over 20,000 documents labeled as contexts. Contexts are selected paragraphs from selected Wikipedia articles. This dataset is widely used for answering tasks, especially in NLP-based research. The SQuAD dataset is a substantial dataset for the task of QA, and Table I presents some details.

TABLE I. DETAILS OF THE SQUAD V2 DATASET

Details	Values
Number of total unique contexts in the dataset:	20,081
Number of total Questions in the dataset:	107,071
Number of total answers in the dataset:	107,071
Maximum questions per context:	50
Minimum questions per context:	1
Average questions per context:	5
Maximum number of words in a context:	653
Minimum number of words in a context:	20
Average number of words in a context:	121

The proposed method for QE uses contextual information given in a sentence using a transformer model such as BERT. For QE, the Masked Language Modeling (MLM) technique was used. Nouns, Adverbs, Proper Nouns, and Adjectives were selected from the given queries and masked. After masking these queries, masked sentences were input into the MLM model to predict the top n tokens for each masked token with their probabilities. Then, the query was expanded by replacing or adding predicted tokens. This expanded query is then used as a question for best context matching, and a transformer model uses the selected context for QA tasks. Figure 1 shows the proposed QA method along with its main components.

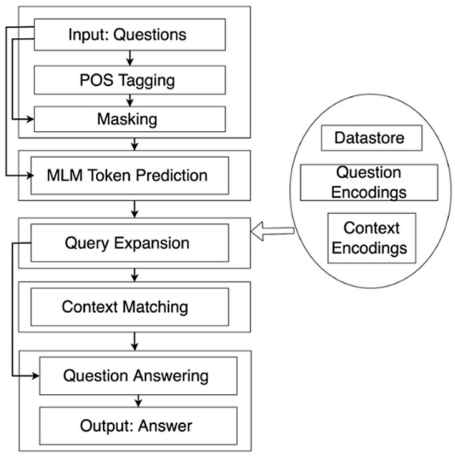

Fig. 1. Overall workflow of the proposed model.

B. System Architecture

1) Masking Module

The first component of the proposed model is a Masking module that takes query sentences as input and performs Parts-Of-Speech (POS) tagging using Spacy (a Python library for various NLP-based tasks). Based on these POS tags, Nouns, Adverbs, Adjectives, and Proper Nouns are replaced with the [MASK] tag. If a Noun is a named entity, then the Masking module will not mask it, as masking named entities can change the overall context of the sentence. For example, in Figure 2, the word Cricket is a proper noun and is a named entity, so the Masking module does not mask it. The Masking module only masks one token/word at a time. These masked query sentences are then passed to the Contextual Token Prediction Module, which predicts the possible alternative tokens that can replace these masked words using the context in the given sentences.

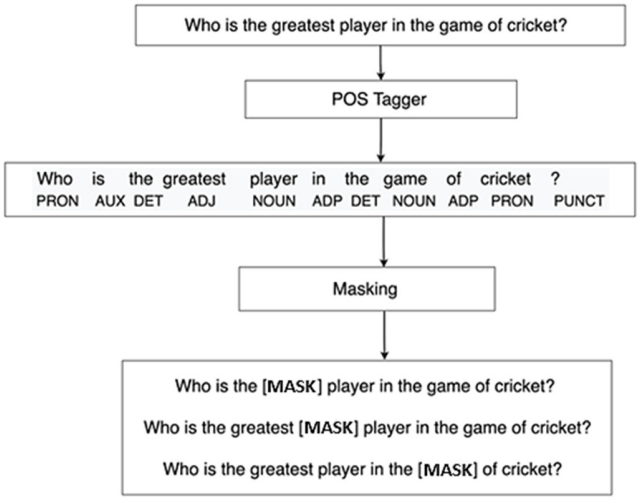

Fig. 2. Example of the Masking module.

2) Token Prediction Module

The Token Prediction module takes each masked sentence/query from the Masking module and predicts possible tokens based on the contextual information in the remaining sentence. Modern, cutting-edge models are inspired by transformers to extract contextual information from given data. The proposed model also uses the Transformer-based fine-tuned BERT model to predict possible tokens that can replace masked tokens in a given masked sentence. The Token Prediction model predicts various tokens, but this method selects only the top n tokens having the highest probabilities. The original sentence/query masked tokens and predicted alternate tokens are then passed to the Query Expansion module to generate an expanded query used in other parts of the model. Figure 3 shows how the Token Prediction module works, predicting masked tokens in sentences masked in the previous step.

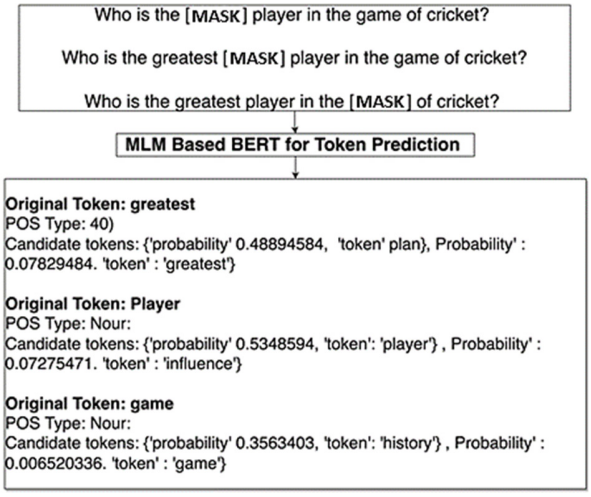

Fig. 3. Example of the Token Prediction module.

3) Query Expansion Module

The Query Expansion module generates or expands the alternative to the original query. It takes the Original Query, Original Masked Tokens, Predicted Tokens, and Probability thresholds as input and returns the expanded query. The alternative token can be replaced in the original token or added as an additional token next to the original in an expanded query. The alternate token will only be appended or replaced in the original query if it is not already present in the original query or its probability is more significant than a certain threshold p. Here, p is the minimum allowed probability of an alternate token, which can range between 0 and 1. Suppose that the token having the highest probability is the same as the original. The next token will be used as an alternate token if its probability is higher than a given threshold. It is advised to utilize a threshold value in the range of 0.3 to 0.6, as in the case of a lower threshold, irrelevant tokens can get replaced. In some cases, if the threshold is too high, the model may not obtain any replaceable token, which will lead to no change in the original query at the time of QE. Figure 4 shows how the Query Expansion module works. In this example, a threshold value $p = 0.4$ is used.

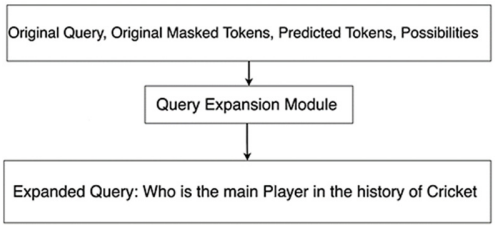

Fig. 4. Example of the Query Expansion Module.

4) Context Extraction Module

This module helps to extract a paragraph or context for the QA task when the context of the question is absent. It is an optional component when the context is already available. This module matches the given questions or expanded questions with already present contexts or questions in a data store. The data store contains the questions and the corresponding contexts from the evaluation set of the SQuAD v2 dataset. For the task of context matching, all contexts and questions were encoded in the data store using the Sentence BERT model, which encodes textual sentences according to their general contextual information. The Context Extraction module extracts relevant context using one of the following two methods. In the first method, a given question (or expanded question) is matched with the questions present in the data store. The cosine similarity between the Sentence-BERT encoding of a given question and all other questions in the data store is calculated, and the context corresponding to the most similar question is returned as the best-matched context. In the second method, the encoding of the given question is directly matched with the encodings of all contexts present in the data store, and the most similar context (based on cosine similarity) is returned. The first method works well in cases where similar questions to the given one are already present in the data store, while the second can be more helpful in the case of new or general questions. Figure 5 shows how this module works.

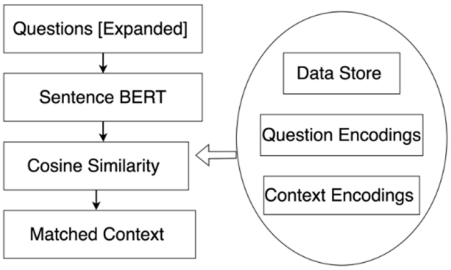

Fig. 5. The Context Matching module.

5) Question Answering (QA) Module

The QA module takes original or expanded questions and a context in which the answer to a given question is present, and provides the response to the query. The context can be a selected paragraph from a document or website, or a matched context extracted from a data store using the Context Matching module.

6) Extracted Most Similar Context for a Query

Here is the most similar context based on an MLM-based expanded query. The document is as follows: "Sachin Tendulkar has been the most complete batsman of his time, the most prolific run maker of all time, and arguably the biggest cricket icon the game has ever known. His batting was based on the purest principles: perfect balance, economy of movement, precision in stroke making, and that intangible quality given only to geniuses' anticipation. If he didn't have a signature stroke-the upright, back-foot punch comes close- it's because he was equally proficient at each of the full range of orthodox shots (and plenty of improvised ones as well) and can pull them out at will." The cosine similarity is 0.6079. The answer is predicted by the QA module using an expanded query, extracting the most similar context. The predicted answer is "Sachin Tendulkar."

IV. EXPERIMENTS

A. Fine-Tuning

Transformer-based BERT models were used for most tasks, including MLM token prediction, document and question encodings, and QA tasks. BERT models have millions of parameters, and many resources and time are required to train them from scratch. Pre-trained models from Huggingface were used, finetuning them to the dataset to avoid this issue.

B. Finetuning the MLM Model

The SQuAD v2 dataset was used to finetune the BERT model for the token prediction task. The same dataset was used to evaluate the QA task. Training the MLM model on the same dataset can help the token prediction model better understand context, while token prediction generates expanded queries. The following operations were performed on the data before passing it to the model for training/fine-tuning.

1) Sentence Extraction

All contexts, questions, and answers were stacked in the SQuAD v2 dataset. As the context is in paragraph format, all contexts were split into sentences. After stacking all the sentences, around 300,000 sentences were obtained.

2) Sentence Filtering

In the sentence extraction process, many useless sentences were selected, such as one- or two-word answers, null sentences, and too-long sentences. Such sentences were filtered out. A minimum and maximum threshold was used to optimally select the minimum and maximum lengths of sentences that can be useful to train an MLM model. Stop words were not removed from sentences, as they help model context understanding. Finally, around 208,000 sentences were obtained after filtering out around 90,000 useless sentences.

3) Tokenization

After selecting usable sentences, they were tokenized using a pre-trained BERT tokenizer. Tokenization is the process of assigning IDs to each word in a sentence according to a predefined vocabulary. Some extra tokens were also added to the process. To standardize the tokens of all sentences, padding tokens '0' were added at the end of the sentence tokens if the length of the sentences was less than the specified maximum length. The tokenizer also returned attention masks with binary values: 1 denotes that an actual token is present, and 0 denotes a padded sentence. All the tokens are attention masks and then stored as tensors in memory.

C. Masking

In this step, a random mask was created having probabilities between 0 and 1. The random mask has the same size as the tokenized tensor. Selection criteria β were defined such that if the random value is greater than the selection criteria model, the original token will be retained; otherwise, this token will be replaced with a masking token. When a small value for the selection criteria is selected, many masked tokens may result in less contextual learning of the model. The masked tokens of all sentences were split into training and evaluation sets.

D. Model Training (Finetuning)

After splitting the data into training and testing sets, the pre-trained BERT model was finetuned. The masked sentence tokens were input into the model, along with the attention masks. The actual sentences were used as target sentences to calculate the prediction loss. The SQuAD v2 dataset was also used to finetune the QA model, using the following steps:

- Data loading: Training and evaluation subsets of the SQuAD v2 dataset were loaded from GitHub. The dataset has contexts, questions, and their corresponding answers.
- Tokenization: Contexts and questions were tokenized using a pre-trained BERT tokenizer.
- Answer locations in context: The location of each answer in context was determined, marking the location of the starting and ending tokens of the context.
- Model training: A pre-trained BERT framework was fine-tuned for the QA task. Token IDs and context attention masks are questions given as input, and the model predicts the start and end locations between which answers may be present. The actual location of the answer is used to calculate the prediction loss.

E. Evaluations and Results

The proposed model's effectiveness was assessed using queries expanded from the MLM-based method compared to queries expanded using WordNet. In the Wordnet-based QE method, the NLTK library (one of the best for NLP tasks) was used for speech tagging and finding synonyms and hypernyms of required tokens. This method masks sentences similarly to MLM (using POS tags). Then, the synonyms and hypernyms of these words were found. Afterwards, two types of expanded queries were generated, one by replacing the original selected words with synonyms extracted from NLTK for that word (selecting the first synonym from the given retrieved set of synonyms), and the other by replacing original selected words with hypernyms extracted from NLTK for that word (the first hypernym from the given retrieved set of hypernyms was selected). After generating these alternate queries, everything else remained the same as for the generated queries using the MLM-based technique.

TABLE II. HYPERPARAMETERS USED IN THIS WORK

Parameter	Value
Probability value for token selection	0.5
Query expansion method	Replace
Minimum threshold for sentence selection	5
Maximum threshold	30
Probability for masking	0.15
BERT model	Bert-base-uncased
Sentence BERT model	Paraphrase-mpnet-base-v2
Model optimizer	AdamW
Programming language	Python
Deep learning library	Pytorch
Compute engine	Colab (GPU-based runtime)
Storage	Google Drive

F. Evaluation Setup

Table II describes different hyperparameters and additional information to reproduce this work.

1) Evaluation of the QA Task

The results obtained from the QA task were evaluated using MLM-based expanded queries and results obtained from expanded queries using other methods. First, all questions in the evaluation set of the SQuAD v2 dataset were expanded using the MLM method for QE. The expanded questions and context corresponding to the original question were passed to the QA model, and then, both types of answers were passed to the same function to make predicted answers and actual answers in the same format. Then, Exact Match (EM) and F1 score were calculated by comparing the predicted answer and the actual one. Similarly, the EM and F1 scores were calculated for the QA model using expanded questions from other query expansion methods. Table III displays the results for the questions expanded from different QE methods on the QA task.

TABLE III. RESULTS FOR THE QA TASK

Method	Matching criteria	F1	EM
Wordnet QE (Synonyms)	QA	68.18	53.61
Wordnet QE (Hypernyms)	QA	55.56	41.32
MLM QE	QA	70.03	55.51

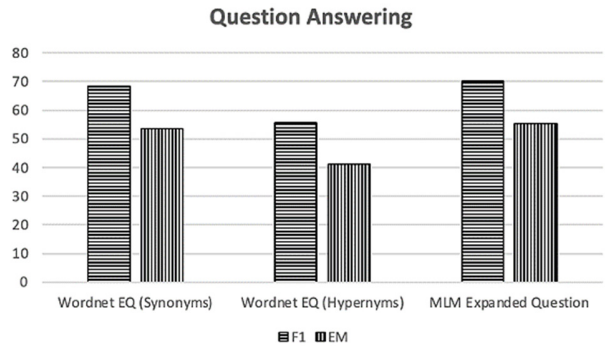

Fig. 6. Results obtained from the QA task.

The proposed model had the best results in terms of both F1 and EM scores, as also shown in Figure 6. In terms of F1-score, the results using the MLM-based QE technique were 2% better than the WordNet synonym-based QE technique and 14% better than the WordNet hypernym-based QE technique. Similarly, EM results using the MLM-based QE technique were 2% better than WordNet synonym-based QE and 15% better than the WordNet hypernym-based QE.

2) Evaluation Criteria for Context Matching

Context-matching is similar to a document ranking task. A question was obtained, and the relevant contexts were ranked according to the similarity index. In this work, to evaluate context matching using different types of queries, all questions were first expanded in the test set of the SQuAD v2 dataset using the proposed MLM-based QE method. Then, context matching was performed using two methods: question matching and direct context matching. In the first method, all original questions were encoded using Sentence BERT and stored. Then, expanded questions were encoded using the same model, and the cosine similarity between the stored and expanded questions was calculated. The corresponding context to the most similar question was then matched with the corresponding context of the original question. The results obtained from context matching using expanded questions using the MLM QE method and other query expansion methods were compared using Mean Average Precision (MAP), Mean Reciprocal Rank (MRR), and F1 score, as shown in Table IV.

TABLE IV. RESULTS ON CONTEXT MATCHING USING THE QUESTION-QUESTION MATCHING METHOD

Method	Match criteria	F1	MRR	MAP
Wordnet QE (Synonyms)	Question-Question	98.54	97.85	92.94
Wordnet QE (Hypernyms)	Question-Question	90.02	85.66	82.01
MLM QE	Question-Question	99.27	99	94.01

The proposed model had the best results in all evaluation metrics. In terms of F1-score, the MLM-based QE technique was 1% better than WordNet synonym-based QE and 9% better than WordNet hypernym-based QE. In terms of MRR, the proposed MLM-based QE technique was 1% better than WordNet synonym-based QE and 15% better than WordNet hypernym-based QE. Similarly, for MAP, the MLM-based QE technique was 1% better than WordNet synonym-based QE and 12% better than WordNet hypernym-based QE.

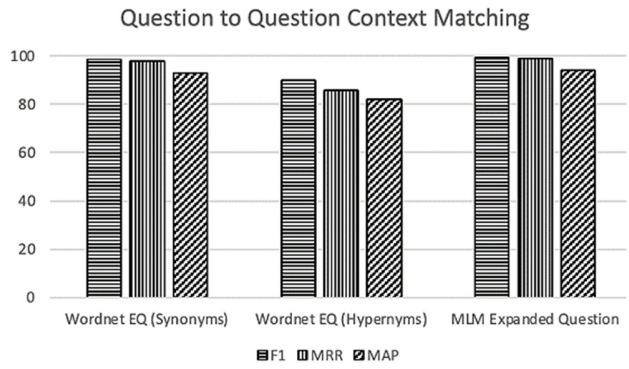

Fig. 7. Results on Context Matching using Question-Question matching methods.

In the second method, all contexts in the dataset were encoded using the Sentence BERT model and stored. The benefit of Sentence BERT is that it encodes data based on the contextual information contained. The expanded question was then encoded using the same model, and the cosine similarity was calculated. The most similar context was selected based on similarity and compared with the corresponding context to the original question for evaluation. Table V displays the results from questions expanded from different QE methods on Context Matching using Question-Context matching. In all evaluation metrics, the proposed model achieved the highest performance. In terms of F1 score, the results using MLM-based QE were 2% better than WordNet synonym-based QE and 18% better than WordNet hypernym-based QE. For MRR, the results of the MLM-based QE were 2% better than WordNet synonym-based QE and 20% better than WordNet hypernym-based QE. Similarly, MAP results using MLM-based QE were 2% better than WordNet synonym-based QE and 20% better than WordNet hypernym-based QE.

TABLE V. RESULTS OF CONTEXT MATCHING USING THE QUESTION-CONTEXT MATCHING METHOD

Method	Match criteria	F1	MRR	MAP
Wordnet EQ (Synonyms)	Question-Context	70.59	64.42	64.42
Wordnet EQ (Hypernyms)	Question-Context	54.2	46.78	46.78
MLM Expanded Question	Question-Context	72.46	66.53	66.53

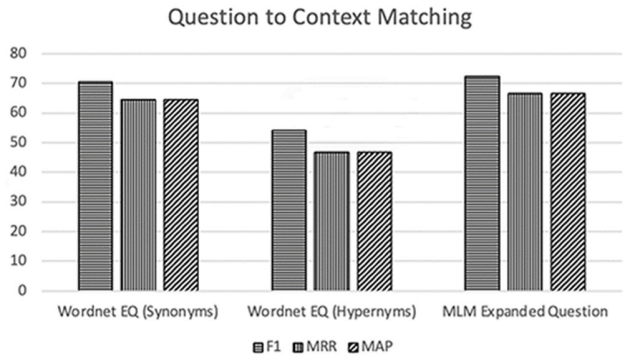

Fig. 8. Results on Context Matching using the Question-Context matching method.

3) Comparison of QE Methods

Table VI compares original questions and their expanded versions and answers generated using these expanded versions. In most cases, it can be observed that queries expanded using WordNet (using synonyms or hypernyms) return wrong answers. In contrast, queries expanded using the proposed MLM-BERT give correct answers to user questions. Questions expanded using the proposed MLM-based technique make more sense. For example, in the second query, expanded versions of questions using both WordNet-based techniques make no sense in answering the question correctly. On the other hand, questions expanded using the proposed MLM-based technique have correct answers.

TABLE VI. COMPARISON OF EXPANDED QUERIES USING DIFFERENT QE METHODS AND PREDICTED ANSWERS BY THE QA MODEL

Query	Expansion method	Original query	Expanded query	Answer
1	Wordnet (Synonyms)	What was the name of Beyoncé's second solo album?	What was the name of Beyoncé's second single album?	Album
1	Wordnet (Hypernyms)	What was the name of Beyoncé's second solo album?	What was the name of Beyoncé's second activity medium album?	Destiny's Child
1	MLM	What was the name of Beyoncé's second solo album?	What was the title of Beyoncé's second studio effort?	B-Day
2	Wordnet (Synonyms)	Beyoncé's father worked as a sales manager for what company?	Beyoncé's father grows_director as a company director for what company?	Topshop
2	Wordnet (Hypernyms)	Beyoncé's father worked as a sales manager for what company?	Beyoncé's parent succeeded income as a administrator institution for what company?	Topshop
2	MLM	Beyoncé's father worked as a sales manager for what company?	Beyoncé's father worked as a campaign manager for what company?	Xerox
3	Wordnet (Synonyms)	In what year did Beyoncé's father quit his job to manage her group?	In what year did Beyoncé's father withdraw his occupation to group her?	2010
3	Wordnet (Hypernyms)	In what year did Beyoncé's father quit his job to manage her group?	In what time period did Beyoncé's parent abdicate her group?	2010
3	MLM	In what year did Beyoncé's father quit his job to manage her group?	In what year did Beyoncé's father quit his job to manage her career?	1995
4	Wordnet (Synonyms)	What is Beyoncé's fan base called?	What is Beyoncé fan base name?	Bey Hive
4	Wordnet (Hypernyms)	What is Beyoncé's fan base called?	What is Beyoncé follower family, installation label?	House of Dereon
4	MLM	What is Beyoncé's fan base called?	What is Beyoncé's fandom called?	Bey Hive

V. CONCLUSION AND FUTURE WORK

This study presented an MLM-based QE mechanism to expand questions and generate related alternate questions for QA tasks. From the above experiments, it can be concluded that the questions expanded using the proposed MLM-based QE method result in relatively more helpful context-matching QA tasks than those expanded using WordNet-based methods. The proposed QE technique offers the best results for context matching and QA tasks in all evaluation metrics. Qualitative analysis also shows that MLM-based expanded questions make more sense, have a meaning similar to the original question, and result in more accurate answers. Although the results of the proposed method are innovative compared to earlier ones, there is always room for improvement. In the future, the QA module can be trained on original questions and expanded versions of them, helping it to learn diverse questions that have the same answers. However, this will require some manual data processing steps, as some expanded versions will need to be discarded before the QE model starts learning them.

FUNDING STATEMENT

This research work was funded by Umm Al-Qura University, Saudi Arabia under grant number: 25UQU4310136GSSR02.

REFERENCES

[1] W. Zheng, H. Cheng, J. X. Yu, L. Zou, and K. Zhao, "Interactive natural language question answering over knowledge graphs," *Information Sciences*, vol. 481, pp. 141–159, May 2019, https://doi.org/10.1016/j.ins.2018.12.032.

[2] O. Kolomiyets and M. F. Moens, "A survey on question answering technology from an information retrieval perspective," *Information Sciences*, vol. 181, no. 24, pp. 5412–5434, Dec. 2011, https://doi.org/10.1016/j.ins.2011.07.047.

[3] T. Hao, W. Xie, Q. Wu, H. Weng, and Y. Qu, "Leveraging question target word features through semantic relation expansion for answer type classification," *Knowledge-Based Systems*, vol. 133, pp. 43–52, Oct. 2017, https://doi.org/10.1016/j.knosys.2017.06.030.

[4] H. Toba, Z. Y. Ming, M. Adriani, and T. S. Chua, "Discovering high quality answers in community question answering archives using a hierarchy of classifiers," *Information Sciences*, vol. 261, pp. 101–115, Mar. 2014, https://doi.org/10.1016/j.ins.2013.10.030.

[5] B. Cabaleiro, A. Peñas, and S. Manandhar, "Grounding proposition stores for question answering over linked data," *Knowledge-Based Systems*, vol. 128, pp. 34–42, July 2017, https://doi.org/10.1016/j.knosys.2017.04.016.

[6] H. J. Oh, S. H. Myaeng, and M. G. Jang, "Semantic passage segmentation based on sentence topics for question answering," *Information Sciences*, vol. 177, no. 18, pp. 3696–3717, Sept. 2007, https://doi.org/10.1016/j.ins.2007.02.038.

[7] Z. Yan, N. Duan, J. Bao, P. Chen, M. Zhou, and Z. Li, "Response selection from unstructured documents for human-computer conversation systems," *Knowledge-Based Systems*, vol. 142, pp. 149–159, Feb. 2018, https://doi.org/10.1016/j.knosys.2017.11.033.

[8] W. Wei *et al.*, "Exploring heterogeneous features for query-focused summarization of categorized community answers," *Information Sciences*, vol. 330, pp. 403–423, Feb. 2016, https://doi.org/10.1016/j.ins.2015.10.024.

[9] A. G. Tapeh and M. Rahgozar, "A knowledge-based question answering system for B2C eCommerce," *Knowledge-Based Systems*, vol. 21, no. 8, pp. 946–950, Dec. 2008, https://doi.org/10.1016/j.knosys.2008.04.005.

[10] F. Wang, W. Wu, Z. Li, and M. Zhou, "Named entity disambiguation for questions in community question answering," *Knowledge-Based Systems*, vol. 126, pp. 68–77, June 2017, https://doi.org/10.1016/j.knosys.2017.03.017.

[11] S. J. Yen, Y. C. Wu, J. C. Yang, Y. S. Lee, C. J. Lee, and J. J. Liu, "A support vector machine-based context-ranking model for question answering," *Information Sciences*, vol. 224, pp. 77–87, Mar. 2013, https://doi.org/10.1016/j.ins.2012.10.014.

[12] A. Rodrigo and A. Peñas, "A study about the future evaluation of Question-Answering systems," *Knowledge-Based Systems*, vol. 137, pp. 83–93, Dec. 2017, https://doi.org/10.1016/j.knosys.2017.09.015.

[13] B. Selvaretnam and M. Belkhatir, "Natural language technology and query expansion: issues, state-of-the-art and perspectives," *Journal of Intelligent Information Systems*, vol. 38, no. 3, pp. 709–740, June 2012, https://doi.org/10.1007/s10844-011-0174-3.

[14] M. Habibi, P. Mahdabi, and A. Popescu-Belis, "Question answering in conversations: Query refinement using contextual and semantic information," *Data & Knowledge Engineering*, vol. 106, pp. 38–51, Nov. 2016, https://doi.org/10.1016/j.datak.2016.06.003.

[15] S. Momtazi and D. Klakow, "Bridging the vocabulary gap between questions and answer sentences," *Information Processing & Management*, vol. 51, no. 5, pp. 595–615, Sept. 2015, https://doi.org/10.1016/j.ipm.2015.04.005.

[16] Y. Gupta and A. Saini, "A novel Fuzzy-PSO term weighting automatic query expansion approach using combined semantic filtering," *Knowledge-Based Systems*, vol. 136, pp. 97–120, Nov. 2017, https://doi.org/10.1016/j.knosys.2017.09.004.

[17] F. Diaz, B. Mitra, and N. Craswell, "Query Expansion with Locally-Trained Word Embeddings," presented at the Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, June 2016.

[18] A. Figueroa, "Automatically generating effective search queries directly from community question-answering questions for finding related questions," *Expert Systems with Applications*, vol. 77, pp. 11–19, July 2017, https://doi.org/10.1016/j.eswa.2017.01.041.

[19] F. C. Fernández-Reyes, J. Hermosillo-Valadez, and M. Montes-y-Gómez, "A Prospect-Guided global query expansion strategy using word embeddings," *Information Processing & Management*, vol. 54, no. 1, pp. 1–13, Jan. 2018, https://doi.org/10.1016/j.ipm.2017.09.001.

[20] P. Azevedo, B. Leite, H. L. Cardoso, D. C. Silva, and L. P. Reis, "Exploring NLP and Information Extraction to Jointly Address Question Generation and Answering," in *Artificial Intelligence Applications and Innovations*, 2020, pp. 396–407, https://doi.org/10.1007/978-3-030-49186-4_33.

[21] K. Karpagam and A. Saradha, "A framework for intelligent question answering system using semantic context-specific document clustering and Wordnet," *Sādhanā*, vol. 44, no. 3, Feb. 2019, Art. no. 62, https://doi.org/10.1007/s12046-018-1022-8.

[22] V. Dibia, "NeuralQA: A Usable Library for Question Answering (Contextual Query Expansion + BERT) on Large Datasets," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Online, July 2020, pp. 15–22, https://doi.org/10.18653/v1/2020.emnlp-demos.3.

[23] M. Esposito, E. Damiano, A. Minutolo, G. De Pietro, and H. Fujita, "Hybrid query expansion using lexical resources and word embeddings for sentence retrieval in question answering," *Information Sciences*, vol. 514, pp. 88–105, Apr. 2020, https://doi.org/10.1016/j.ins.2019.12.002.

[24] E. Damiano, A. Minutolo, S. Silvestri, and M. Esposito, "Query Expansion Based on WordNet and Word2vec for Italian Question Answering Systems," in *Advances on P2P, Parallel, Grid, Cloud and Internet Computing*, 2018, pp. 301–313, https://doi.org/10.1007/978-3-319-69835-9_29.

[25] I. Lahbari, S. E. Alaoui, and K. Zidani, "Toward a New Arabic Question Answering System," *The International Arab Journal of Information Technology*, vol. 15, no. 3A, pp. 610–619, 2018.

[26] W. Bakari, P. Bellot, and M. Neji, "A logical representation of Arabic questions toward automatic passage extraction from the Web," *International Journal of Speech Technology*, vol. 20, no. 2, pp. 339–353, June 2017, https://doi.org/10.1007/s10772-017-9411-7.

[27] H. K. Azad and A. Deepak, "A new approach for query expansion using Wikipedia and WordNet," *Information Sciences*, vol. 492, pp. 147–163, Aug. 2019, https://doi.org/10.1016/j.ins.2019.04.019.

[28] S. Kandasamy and A. K. Cherukuri, "Query expansion using named entity disambiguation for a question-answering system," *Concurrency and Computation: Practice and Experience*, vol. 32, no. 4, 2020, Art. no. e5119, https://doi.org/10.1002/cpe.5119.

[29] D. Parapar, A. Barreiro, and D. E. Losada, "Query expansion using wordnet with a logical model of information retrieval," in *IADIS AC*, 2005, pp. 487–494.

[30] R. Chauhan, R. Goudar, R. Rathore, P. Singh, and S. Rao, "Ontology Based Automatic Query Expansion for Semantic Information Retrieval in Sports Domain," in *Eco-friendly Computing and Communication Systems*, 2012, pp. 422–433, https://doi.org/10.1007/978-3-642-32112-2_49.

[31] P. Sharma and N. Joshi, "Knowledge-Based Method for Word Sense Disambiguation by Using Hindi WordNet," *Engineering, Technology & Applied Science Research*, vol. 9, no. 2, pp. 3985–3989, Apr. 2019, https://doi.org/10.48084/etasr.2596.

[32] D. Chopra, N. Joshi, and I. Mathur, "Improving Translation Quality By Using Ensemble Approach," *Engineering, Technology & Applied Science Research*, vol. 8, no. 6, pp. 3512–3514, Dec. 2018, https://doi.org/10.48084/etasr.2269.

[33] B. Nethravathi, G. Amitha, A. Saruka, T. P. Bharath, and S. Suyagya, "Structuring Natural Language to Query Language: A Review," *Engineering, Technology & Applied Science Research*, vol. 10, no. 6, pp. 6521–6525, Dec. 2020, https://doi.org/10.48084/etasr.3873.

[34] P. Rajpurkar, R. Jia, and P. Liang, "Know What You Don't Know: Unanswerable Questions for SQuAD." arXiv, June 11, 2018, https://doi.org/10.48550/arXiv.1806.03822.