International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Revue d'Intelligence Artificielle
Vol. 37, No. 6, December, 2023, pp. 1577-1585
Journal homepage: http://iieta.org/journals/ria

Combined Approach for Answer Identification with Small Sized Reading Comprehension Datasets

Pradnya S. Gotmare*, Manish M. Potey

Department of Computer Engineering, K J Somaiya College of Engineering, Vidyavihar University, Mumbai 400077, India

Corresponding Author Email: pradnyagothmare@somaiya.edu

https://doi.org/10.18280/ria.370622

Received: 27 July 2023
Revised: 7 September 2023
Accepted: 30 October 2023
Available online: 27 December 2023

Keywords:
annotations, embedding, semantic, comprehension, transformers

ABSTRACT

In the realm of natural language understanding, machine reading and comprehension have emerged as significant areas of interest, requiring machines to extract pertinent information from textual data and understand it. This study proposes a novel method for answer identification in a multiple-choice question answering setup, utilizing science textbook and narrative text data. The proposed methodology integrates lexical semantic features at the word level and sentence-level equivalence. Initially, the strategy exploits lexical features, particularly word overlap, critical for answer identification. It extracts features such as noun phrases, verb phrases, and prepositions, accounting for their grammatical relationships. These features are then enhanced by assessing semantic similarity via a transformer model. Subsequently, answer identification is executed by mapping between answer option sentences and paragraph sentences on a one-to-one basis. The accuracy of correct answer identification was evaluated using both a feature-based approach and a BERT-based approach. Results indicated an accuracy of 66.4% and 57.5% for the science and narrative datasets, respectively, employing the combined approach. The performance evaluation of the proposed method was undertaken with a fine-tuned pre-trained language model. The evaluation analysis revealed certain challenges with the proposed methodology, outlining avenues for future research.

1. INTRODUCTION

Machine reading and comprehension framework requires understanding of the given paragraph text and answering the questions based on the given paragraph text. In this work input consists of paragraph P, questions based on paragraph (qi), four answer options (ai) and one correct answer option. With the given input, system identifies the correct answer option out of the four given options. This work emphasizes role of pre-trained language tools such as POS tagger, dependency graph and pre-trained language models like Sentence-BERT, to identify the answers from the paragraph text data.

It is very apparent that machine need to understand natural language text to extract the relevant part of the passage and provide answer. In AI based systems, it has been challenging task to extract relevant information from textbook data. In Machine Reading and comprehension (MRC) systems information can be represented as: {{P}, {Q}, {A}} where P={s1, s2, s3, ..., sn} have number of sentences (s_i) of the paragraph P; Q={q1, q2, q3, ..., qn} represents questions based on paragraphs and A={a1, a2, a3, a4} where a_i is an answer option.

Hypothesis: Given a paragraph text, question qi, answer options ai, system identifies the correct answer option if combination of q+ai matches with relevant sentence/sentences of the given Paragraph. Set P={S1, S2, …, Sn}, is represented as a set of sentences. Set Q={q1, q2, ...}, indicates no of questions on paragraph. Set A={a1, a2, a3, a4} indicates, four possible answer options, out of which one of the answer option is correct. Answer option sentence (q+ai) is obtained by combining question with answer option.

Lexical similarity indicates matching of extractive features. Noun, verb, preposition and certain combinations like noun phrase chunk, verb-preposition (v-prep), noun with preposition (n-prep), are considered as extractive features.

Meaningful grouping of words on the basis of syntax is indicated by noun phrase chunks, and verb phrase chunks. Such meaningful grouping of words indicate lexical semantic feature. These features are identified in answer option sentence and paragraph sentence. These features are obtained by using POS Tagger and dependency graph by executing Stanford’s annotation pipeline.

Semantic similarity indicates the equivalence of sentences based on mainly noun phrases, verb phrases and prepositions. It is achieved by using sentence embeddings and cosine similarity feature.

BERT (Bidirectional Encoder Representations from Transformers) is a transformer based model, that is pre-trained on large corpora of text data. Sentence-BERT (SBERT) is a modification of the pre-trained BERT network that uses siemese triplet network structure to derive semantically meaningful sentence embeddings [1].

SBERT (sentence-BERT) transformer is used to identify semantic textual similarity between pair of sentences, namely answer option sentences and paragraph sentences.

SBERT generates semantically meaningful sentence embeddings. These embeddings are compared using cosine similarity. This feature is used to identify the equivalent answer option sentence with paragraph sentence.

Various features of lexical and semantic similarity are taken into consideration in combination with SBERT sentence embeddings.

The proposed approach is the combination of knowledge base and usage of transformer model for sentence embedding. Knowledge base is constructed by considering the dependency graph returned by Stanford CoreNLP parser. Every sentence is represented as a sequence of tokens. In the dependency graph every token is connected with other token having certain grammatical relationship. From this grammatical dependency relationships specific combinations of patterns can be generated.

Knowledge graph is an another representation. It is an abstract graph that consists of nodes corresponding to entity and edges corresponding to specific grammatical relationships. In this representation, noun and noun phrases are the entity nodes which are connected with edges corresponding to grammatical relationships like verb, preposition and verb-preposition edges. It is needed to identify whether a verb edge exists among two noun/noun phrase chunks?, or is there a verb-preposition edge? Knowledge graph is generated from the dependency relations of noun, verb, and preposition provided by dependency graph. This knowledge graph is stored as a knowledge base.

In the recent work in reading comprehension most of the systems are based on deep learning approaches [2-6] which need huge amount of input data. The proposed approach is for small sized textual data which is not sufficient for deep learning. Another significance of the work is to identify the framework which can be generalized for different domains/genres. Dataset of science textbook and stories are completely different kinds of genres. There is a need of common framework that can be applied to machine reading comprehension system of different genres having small dataset.

Recent state of the art BERT models helps in locating the correct answer based on the context, but these models donot perform well for certain compound words/phrase matching tasks. It has been observed that the compound noun having cardinal values are not considered equivalent by SBERT model. The sentences having noun phrase chunks like 12-18 years and 12 to 18 years are not identified correctly by SBERT model. In contrast to this, compound nouns with cardinal values are identified as a meaningful noun phrase chunk by shallow parsing with dependency graph. Grammatical features of the text provides a clue for identifying extractive features from the passage text. These features are further augmented with embedding feature provided by SBERT model. Many researchers have emphasized use of popular word embedding models like word2vec, Glove, Fasttext for textual level similarity. In the recent work in sentence embeddings, BERT based transformer models has been found fruitful for identifying sentence level semantic similarity in low resource datasets. In this work we have considered the extractive features alongwith BERT (sentence-BERT) embedding score to identify the answer out of the given options. We have used 'all-MiniLM-L6-V2' transformer model from Huggingface open source Library [7] for obtaining sentence embeddings. This approach is combination of extractive features based on lexical semantics and semantic features based on embeddings.

It has been observed that the proposed methodology is applicable to both the domains. Some of the problems identified with proposed methodology are listed in evaluation analysis. It can be taken as a future research. This methodology can be visualised to demonstrate the stepwise procedure followed in reading comprehension.

This paper is organized in different sections as follows. In section 2, we have described the various features applicable to language understanding applications. Section 3 discusses the proposed methodology based on knowledge base generation and sentence embeddings. Section 4 mentions findings of the experimentation and error analysis. Finally we conclude the challenges in machine reading application and provides future directions.

2. RELATED WORK

Machine learning approaches are mainly used for construction of feature space required in comprehension based systems [8]. In the work [9] author has described task specific retrievers to get relevant contexts at an appropriate level of semantic granularity. Complex QA system [10] describes the different language model architectures, strategies and challenges in terms of task complexity and evaluation. Major features required for token level and sentence level tasks are described in the following section.

2.1 Lexical semantic feature identification

In case of reading comprehension major task is to locate relevant parts of the passage by identifying entities, relations, lexical properties [11, 12] semantic properties [13-18] of the given text. Answer can be extracted directly from the paragraph text or from some intermediate form. While using an intermediate form, sentences of the paragraphs are represented in structured format like database table, semantic graphs or annotated form as an intermediate form. Questions are answered based on such intermediate representation [18]. The most common way of dealing with MRC tasks is to train machine learning model on an annotated database [19]. In case of hybrid form of QA [20], multiview is considered for answering over table and text. In multiview author has explained question answering based on span of text and tabular data. In our work dependency graph is used to extract the span of text which corresponds to noun chunk phrases.

2.2 Significance of word embedding

Pretrained word embeddings are an integral part of modern NLP systems, which offer significant improvements over embeddings learned from scratch [1]. Word embedding is a type of word representation that enables words with similar meaning to have similar representation. In this technique each word is represented by a real valued vector, which has almost hundreds of dimensions. Latent semantic analysis and skip gram are the mostly used methods for learning word vectors. In the recent work [21], FastText is considered to identify word similarity related to dilects in Arabic text for opinion analysis. In another work [22, 23] embeddings are created using popular algorithms like word2vec, FastText, Glove (Global vectors for word representation) etc.

In the recent work many authors have described various models using BERT embeddings.

2.3 Role of BERT embedding

In the recent work with BERT embedding [24], author has described textual entailment for classification for legal text documents. In this application, author has mentioned the use of sentence BERT model along with metadata of the civil court for entailment classification. Author [25] has described need of pretrained models with BERT for the task of retrieval and classification of scientific abstracts. In another work [26] author has used BERT's transfer learning ability for enhancing performance of decision making in sentiment analysis. In this work, authors have also compared popular word embedding techniques such as Glove, Fast Text and Word2Vec with BERT. Combination of improved BERT model (iBERT) [27] and dependency trees are used to construct semantic representation of the text in sentiment analysis. In case of BERT-based method (BERT-ConvE) [28], embeddings are used to represent node text attributes to complete the knowledge graph.

2.4 S-BERT for sentence embedding

Sentence-BERT (SBERT) is a modification of the pre-trained BERT network that uses siemese triplet network structure to derive semantically meaningful sentence embeddings [1]. BERT’s model architecture is a multilayer bidirectional transformer encoder based on original implementations of attention [30] as shown in Figure 1 and Figure 2. The Biencoder produces embeddings for the paragraph sentences as well as for the answer option sentences. These embeddings are produced independently. SBERT model enhances the BERT model by adding a pooling operation to its output. It is shown by Figure 3.

In this architecture sentence A correspond to paragraph sentences and sentence B corresponds to answer option sentences. U and V represents sentence embeddings.

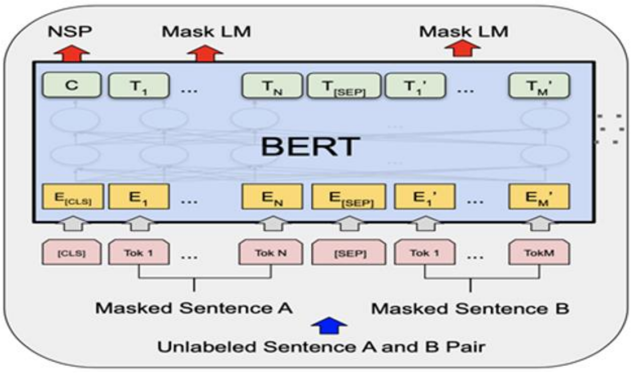


Figure 1. Overall BERT architecture [29]

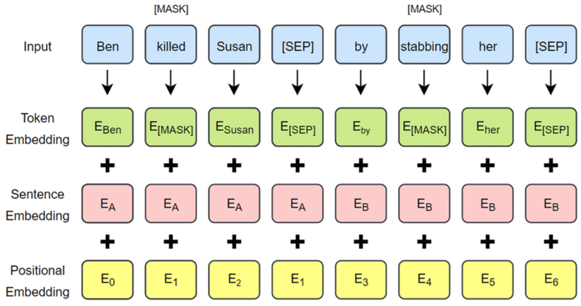


Figure 2. BERT input embeddings [29]

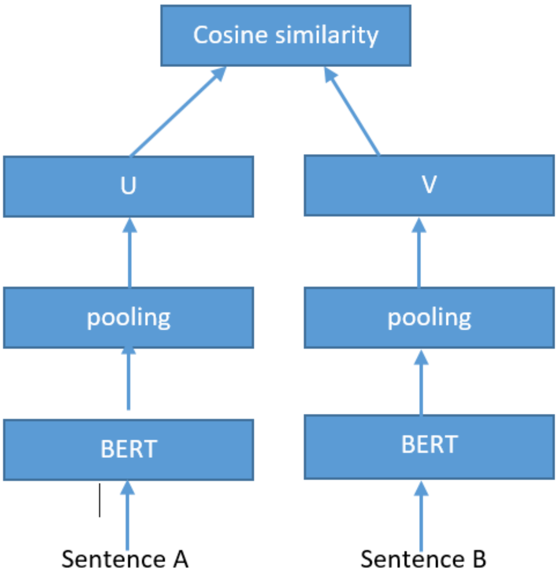
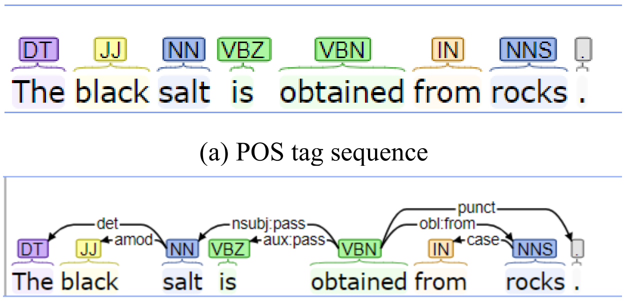


Figure 3. SBERT architecture with role of Biencoder [1]

2.5 Dependency parsing and knowledge generation

In an abstract form, sentences of a paragraph are represented with different graph structures such as sequence graph and dependency graph [31]. Stanford CoreNLP parser provides the annotation pipeline where sentences are represented with different grammatical relationships. Shallow parsing with enhanced dependency parse provides semantic information associated with textual data such as noun phrase/chunks, verbs, preposition phrases, clauses etc. Grammatical features extracted from dependency parser provide semantic information, needed for relevant textual information. This semantic information can be represented in a structured form like a knowledge graph and predicate argument structure [32].

The Stanford CoreNLP library provides API’s which can perform different text operations for natural language processing like parsing, tokenization, lemmatization, parts of speech tagging, chunking, sentence segmentation, Named Entity Identification and coreference identification. NLTK, OpenNLP, Spacy Toolkits are also available to build more advanced text processing services for processing of natural language text. Figure 4 shows representation of sequence graph and dependency graph obtained using CoreNLP parser.


DT
JJ
NN
VBZ
VBN
IN
NNS
The black salt is obtained from rocks .
det
amod
nsubj:pass
aux:pass
punct
obl:from
case
DT
JJ
NN
VBZ
VBN
IN
NNS
The black salt is obtained from rocks .

(a) POS tag sequence

(b) Dependency graph

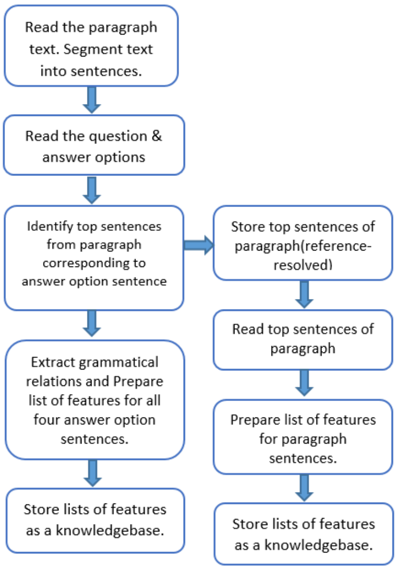
Figure 4. POS tags and dependency graph from CoreNLP [33]

Sentence: The black salt is obtained from rocks.

3. METHODOLOGY

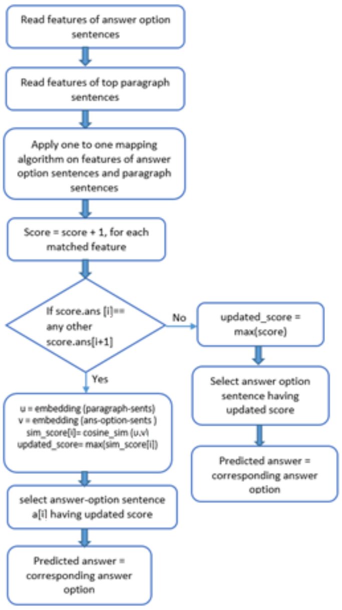
Question answering is the task of identifying answer for the question based on the support text. Most of the comprehension based answering systems locate the correct answer in the given paragraph by identifying proximity with the question words [34]. In the proposed comprehension based QA system, textual data is given in the form of paragraph sentences. Questions are given in the form of multiple choice options. Answer is the relevant sentence of paragraph matching with the answer option sentence. Answer option sentence is formed by combining multiple choice option with the given question as mentioned in section 3.1.

The problem of answer extraction is treated as an optimal subgraph identification in the given paragraph text. In order to get the optimal subgraph, the three major predicates such as noun phrase chunks, verb and preposition have taken into consideration. The detailed methodology is described with Figure 5 and Figure 6.



Read the paragraph text. Segment text into sentences.
Read the question & answer options
Identify top sentences from paragraph corresponding to answer option sentence
Store top sentences of paragraph(reference-resolved)
Read top sentences of paragraph
Extract grammatical relations and Prepare list of features for all four answer option sentences.
Prepare list of features for paragraph sentences.
Store lists of features as a knowledgebase.
Store lists of features as a knowledgebase.

Figure 5. Methodology_phase-1



Read features of answer option sentences
Read features of top paragraph sentences
Apply one to one mapping algorithm on features of answer option sentences and paragraph sentences
Score = score + 1, for each matched feature
If score.ans [i]== any other score.ans[i+1]
No
Yes
updated_score = max(score)
Select answer option sentence having updated score
Predicted answer = corresponding answer option
u = embedding (paragraph-sents)
v = embedding (ans-option-sents)
sim_score[i]= cosine_sim (u,v)
updated_score= max(sim_score[i])
select answer-option sentence a[i] having updated score
Predicted answer = corresponding answer option

Figure 6. Methodology_phase-2

3.1 Construction of answer option sentence

In the given setup four options are given for every question. The multiple choice answer options are in the form of a single word, phrase or a brief sentence. Answer option sentence is formed by combining question with the given answer option by using patterns mentioned with specific formats [35]. The question like Which mineral makes strong bones? is replaced with Which +V.<makes>.(+NP).

It can be written as NP +V. <makes>. (+NP), where 'which' word can be replaced with given option represented as a noun phrase (NP).

Answer option sentences are formed by crowd workers using the pattern identification and rewriting rules as mentioned above. This task is done in offline mode. There are different types of pattern rewriting rules [35]. Answer option sentences are the framed answers obtained from given answer options.

3.2 Top sentences identification and storage

There are few sentences of paragraph which match with the answer sentences. It means answer of the question lies in specific cluster of sentences. Initially Top paragraph sentences are identified by considering cosine similarity feature among answer option sentence embeddings and paragraph sentence embeddings. This Pairwise sentence scoring task is performed as shown in Algorithm 1. Identified Top sentences are stored in the form of lists. This algorithm is applied to reduce the search space.

Algorithm 1: The use of cosine similarity feature among two embeddings

```
Algorithm 1: To get top sentences for each question.
Input: paragraph (Pi), question (qi), answer option sentences.
Output: Lt - List of questionwise top sentences.
for i = 1 to n do // sentences of paragraph
{
    Corpus = paragraphSentence[i]
    corpus-embeddings = embedder.encode(Corpus)
    for j = 1 to 4 do
    {
        Read answer option sentence a[j] for a question
        query-embeddings = embedder.encode (answer option sentence a[j])
    }
    sim_score[i] = cosine_sim(query-embeddings, corpus-embeddings)
    Update Lt with top paragraphSentence[i] using sim_score
}
return Lt.
```

3.3 Generation of knowledge graph

The process of answer extraction from paragraph text needs identification of relationships among the nodes of the graph. It is essential to identify semantic roles of major predicates like noun, verbs, and preposition.

Knowledge graph is an abstract graph that consists of nodes corresponding to entity and edges corresponding to specific grammatical relationships. Noun and noun phrases are the entity nodes which are connected with edges corresponding to grammatical relationships. Depparse pipeline of Stanford

CoreNLP Tool is used to obtain different dependency relationships (dependency associated with noun, verb, and preposition phrases appearing in the sentence). This information is used to generate knowledge base. This knowledge base consists of lists of specific patterns. Knowledge graph is represented as a knowledge base to store specific relationships.

3.4 Extraction and pattern generation phase

Initially every sentence is represented as a sequence of tokens. Every sentence of the paragraph is a connected graph G (V, E) represented as a dependency graph, where V(nodes) are tokens and these tokens are connected with certain grammatical relationship with other tokens. G1(V1, E1) is the graph for every answer option sentence. In order to generate knowledge from these graphs, specific combinations of patterns are considered.

Combination 1:
G (Si) - shallow parse, noun phrase chunks
G1 (Ai) - shallow parse, noun phrase chunks

Combination 1: shows noun phrase chunks obtained with shallow parsing. Noun phrase chunk indicate noun phrase (NP) with premodifiers.

Noun phrase (NP) with premodifiers consists of article, adjectives and noun.

Noun phrase is a meaningful groupings of tokens. With this combination NP of answer option sentence is matched/mapped with NP of paragraph sentence.

Combination 2:
G (Si) - shallow parse, noun phrase, verb
G1 (Ai) - shallow parse, noun phrase, verb

Combination 2: It is observed that whether noun phrase appear along with verb. Verb is a specific relationship which is connected with subject noun and object noun.

Combination 3:
G (Si) - shallow parse, noun phrase, verb, preposition
G1(Ai) - shallow parse, noun phrase, verb, preposition.

Combination 3: it is observed that whether the noun phrase appears with verb and preposition.

where, Si - paragraph sentence. Ai - answer option sentence

In all three combinations noun is a part of noun phrase.

Score function with noun phrases, verbs and prepositions is represented as f(x)=f(x1, x2, x3) such as x1, x2, x3 belongs to Si and Ai.

x1 → noun phrases, x2 → verbs and x3 → prepositions.

f (x) is the score function with x1, x2, x3 that considers one to one mapping between answer option sentence (Ai) and paragraph sentences (Si).

Specific patterns are created from these combinations, those are termed as annotated patterns, e.g. patterns like verb-propositions (vprep means verb followed by preposition).

Created patterns are stored in structured form. The features considered in this setup are noun, noun phrase, prepositions, verbs, adverbs, verb-preposition, noun-preposition, subject-object edge corresponding to verb.

Algorithm 2 and 3 indicate extractive feature identification and storage in the form of lists. The top sentences of paragraph are coreference resolved sentences [36].

Algorithm 2: Prepare lists of features for top sentences of paragraphs

```
Input: Lt - List of top sentences of paragraph, qi.
Output:
L1: list of noun phrases
L2: list of nouns
L3: list of prepositions
L4: list of verbs
L5: list of adverbs (adv)
L6: list of verb-preposition pattern (vp)
L7: list of noun-preposition pattern (np)
L8: list of sub-obj pattern
Output File F1.    // features of top sentences
{
for Si in Lt //sentence of paragraph (Si)
Execute nlp-annotation-pipeline()
Extract features as mentioned in L1, L2, L3, L4, L5, L6, L7,
L8 using POS (part of speech) tags.
Save lists L1, L2, L3, L4, L5, L6, L7, L8 to file F1
Return F1.
```

Algorithm 3: Prepare list of features for answer option sentences

```
Input: File F2    //{qi and list of answer option sentences }
Output: File F3  //{answer option pattern combinations}
while EOF(F2)
for i = 1 to 4 do    //{for answer option sentences}
Execute nlp-annotation-pipeline()
Extract features as mentioned in L1, L2, L3, L4, L5, L6, L7
L8 using POS tags.
Save lists L1, L2, L3, L4, L5, L6, L7, L8 to file F3
Return F3.
```

3.5 Mapping algorithm

One to one mapping of features is applied between an answer option sentences and paragraph sentences. Score is calculated on the basis of matching of features in answer option sentence and paragraph sentence. The answer option sentence having maximum score with paragraph sentence is considered as the correct answer option.

If score of an answer option sentence is equal to any other answer option sentence, then the embedding score of all those answer option sentences is considered. Embedding scores of paragraph sentences are obtained. Maximum cosine similarity score is obtained from both the embeddings. The answer option sentence having maximum cosine similarity score is considered as the correct answer option.

In this setup at first score of extractive features is considered. Cosine similarity score of embeddings is considered if score of extractive features is same with more than one anwer option sentence.

Calculation of score based on extractive features and cosine similarity of embeddings is shown with Algorithm 4.

Algorithm 4: Mapping algorithm using features and sentence embeddings with transformer

```
Input: Files F1, F2, F3
Output: predicted answer sentence.
{
Read F1
score = 0
```

```
with F1
Read features associated with paragraph sentences
with F3
Read features associated with answer option sentences
for i = 1 to 4 do {
  for j = 1 to lengh (Si) where Si → top sentences of paragraph
  {
    noun_score = n (a[i]. noun ∩ S[j].noun)
    nsubj_score = n ( a[i].nsubj ∩ S[j].nsubj) // noun as a subject
    dobj_score = n (a[i].dobj ∩ S[j].dobj) // noun as a object
    verb_score = n (a[i].verb ∩ S[j].verb) // verb
    np_score = n (a[i].nprep ∩ S[j].nprep) // noun-preposition
    vp_score = n (a[i].vprep ∩ S[j].vprep) // verb-preposition
    adv_score = n (a[i].adv ∩ S[j].adv ) //adverb
    adj_score = n (a[i],adj ∩ S[j].adj ) //adjective
    score = noun_score + nsubj_score + dobj_score+ verb_score + np_score + vp_score + adv_score + adj_score
    }
score = max(score)
If (score of a[i] equal to score of any other a[i+1]) then
Call embedding()
Consider a[i] corresponding to updated_score
else
consider a[i] corresponding to max(score)
return a[i]
}

embedding()     // embedding function
{
corpus = Si
corpus-embeddings (u) = embedder.encode(corpus)
i = 0
while (i < 4) {
query-embedding (v) = embedder.encode(a[i])
sim_score[i] = cosine-sim ( u, v )
}
updated_score = max(sim_score[i])
return updated_score
}
```

3.6 Implementation of embedding in score calculation

Two different approaches are used to calculate the score for identifying answer sentence. In the first approach extractive features are considered along with cosine similarity of embeddings as shown in algorithm 4. The option having highest score is considered as an answer.

In the second approach embeddings score is obtained with sentence-BERT without considering extractive features. In this approach cosine similarity score of embedding is the only feature considered for answer identification. This is described with embedding() function.

Combined score is calculated by adding score from approach1 followed by approach2. The answer option sentence having highest score value is considered as an answer. The corresponding option is the predicted answer.

3.7 Combined score calculation using normalisation

Score obtained with extractive features is an integer value while the score obtained with cosine similarity of embeddings is a value in the range of 0 to 1. Euclidean normalization is applied to get the normalized score value from extractive feature score. Normalized_score is in the range of 0 to 1. Combined score is the addition of normalized_score and cosine_simscore. Predicted answer is the option corresponding to maximum value of combined_score.

Normalized_score=Eucledian(Feature_score)
Combined_score=Normalized_score+Cosine_simscore

3.8 Data set under consideration

This dataset comprises paragraphs sourced from elementary and middle school science textbooks, supplemented with multiple-choice questions drawn from competitive examinations. Furthermore, it incorporates science textbook questions from the MultiRC [33] dataset.

Second dataset is MCTest [37], it is also freely available stories data for reading comprehension. It is a dataset about fiction stories at elementary level created by crowd workers. Details of the datasets are as given in Table 1.

Table 1. Datasets

Dataset No.	Dataset Title	No of Paragraphs	No of Questions
Dataset 1	Science textbook	141	310
Dataset 2	MCTest	125	1000

Note: The code and dataset is available at https://github.com/Prad1got/MRC

4. RESULTS AND DISCUSSION

4.1 Experimental results

Accuracy is considered as an evaluation criteria, where each question has one correct answer among four provided options. Accuracy is defined as:

$$accuracy = \frac{predicted\ correct\ answers}{total\ correct\ answers}$$

Type of questions

In case of science textbook dataset majority of the questions fall in the categories like What, Which, Why, How etc. With stories dataset majority of the questions fall in the categories like What, Who, Why, How etc.

Table 2. Dataset 1: Question types and count of predicted correct answers

	What	Why	How	Which	Other	Total
Total Qs	132	23	29	113	13	310
Appr1	78	16	18	68	7	187
Appr2	84	15	16	72	8	195
combined	88	19	17	74	8	206

Table 2 and Table 3 shows statistics about type of questions and count of the correct answers identified with two different approaches. It has been observed that the same methodology is applicable to two distinct domains. With science textbook dataset, accuracy achieved using approach 1 is 60.3% while it is 55% for stories dataset. With approach 2 accuracy is 62.9% and 56% for science dataset and stories dataset respectively. With combined approach, there is an increase in overall accuracy approximately by 6% and 2.5% for science text dataset and stories dataset respectively. With sentence-Bert, there is a slight decrease in the accuracy of nonfactoid questions like why and how.

Appr1 (Approach-1) - with extractive features and embedding feature approach

Appr2 (Approach-2) - with only embedding feature approach.

Combined - Approach 1 followed by Approach 2

Table 3. Dataset 2: Question types and count of predicted correct answers

	What	Why	How	Who	Other	Total
Total Qs	524	132	81	130	133	1000
Appr1	298	74	39	76	63	550
Appr2	305	69	42	82	62	560
Combined	311	76	41	88	59	575

Table 4. Datasets and accuracy

Datasets	Features Used	Total	Predicted Correct Answers	Accuracy
Dataset-1	Appr1	310	187	60.3%
	Appr2		195	62.9%
	Combined		206	66.4%
Dataset-2	Appr1	1000	550	55 %
	Appr2		560	56 %
	Combined		575	57.5%

Accuracy results obtained by implementing the proposed methodology with two different genres, datasets are listed in Table 4. Accuracy of correctly predicted answers is shown with Figures 7 to 10, for significant types of questions in both the datasets.

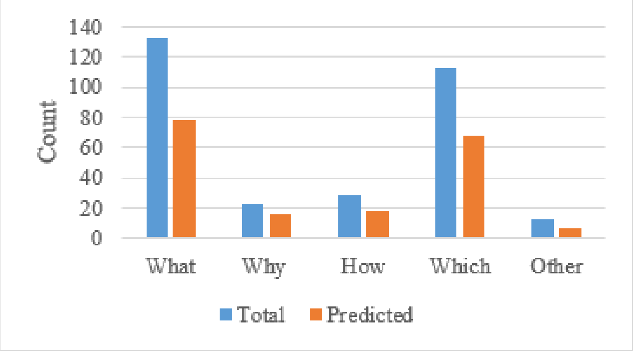


Figure 7. Accuracy using approach1 on Dataset1

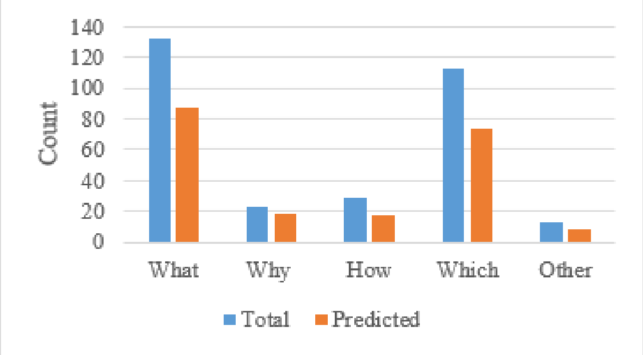


Figure 8. Accuracy using combined approach on Dataset1

Accuracy of answering is dependent on word overlap and phrasal similarity between answer option sentence and probable paragraph sentences. Certain shortcomings of the methodology are identified as below.

(1) Sentence embedding

Sentence embedding is used for identifying equivalent noun phrases and equivalent verbs, but it has certain limitations. More work is needed to increase accuracy at contexual level. The sentences having comma separated nouns appearing in the form of list need specific attention. The different grammatical constituents of a sentence need to be explored to generate knowledge.

(2) Referencing

Referencing is another problem when identifying connected sentences (cluster of sentences) or connected noun phrases appearing in the same sentence or in subsequent sentences.

Accuracy can be increased by identifying correct reference terms and resolving those references. In this setup, coref pipeline of Stanford parser is used for coreference resolution.

Both the datasets have factoid and nonfactoid questions. Whenever the questions or the paragraph sentence include negation, this system does not predict correct answer. Similarly some of the questions are based on sequence of events or processes. Such questions are not predicted correctly by our system. In the present setup there is no provision for handling these cases. Another challenge for the system is questions based on common sense knowledge and implicit reasoning. This can be taken as a future scope.

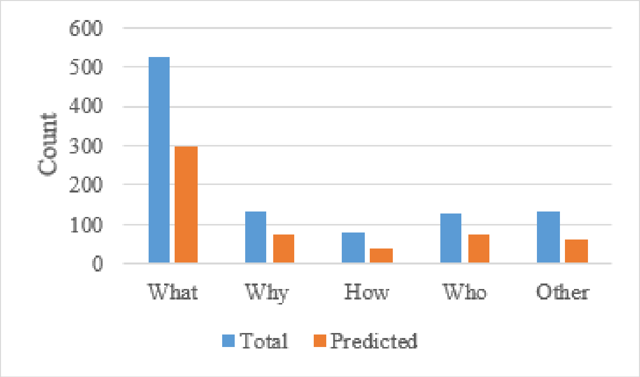


Figure 9. Accuracy using approach1 on Dataset2

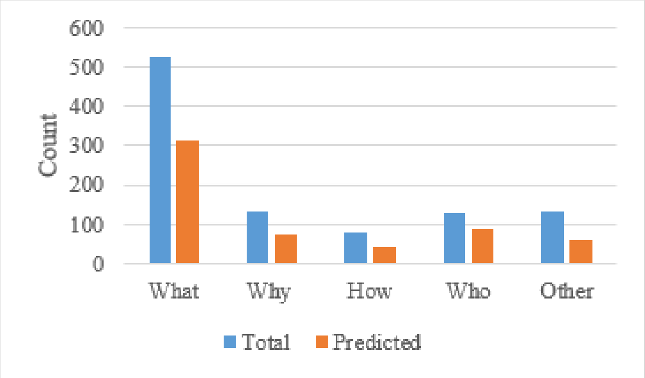


Figure 10. Accuracy using combined approach on Dataset2

4.2 Performance evaluation

ALBERT (A Lite BERT) is a pre-trained model that is widely used architecture in question answering domain for fine-tuning. ALBERT configuration is similar to BERT Large and can be trained about 1.7 x times faster [38]. Evaluation is performed by finetuning pretrained ALBERT [38] model that need data in SQuAD dataset format. SQuAD is a popular format for training and evaluating language models for Question-Answering tasks. SQuAD format includes passage-text, accompanied with question and corresponding answer. We have converted our dataset into SQuAD data format. After that fine tuning of pretrained ALBERT model is performed on subset of our datasets.

For exact match (EM) answer score value is considered as 1 while for partial correct match answer score is considered as 0.5. In case of no matching, answer score value is considered zero. Performance comparison for both the datasets is given in Table 5.

Table 5. Performance evaluation with pretrained Q-A model

Dataset No.	Dataset Title	Albert (Pretrained LM)	Proposed Approach
Dataset 1	Science textbook	52.8 %	66.4%
Dataset 2	MCTest	44.5%	57.5%

5. CONCLUSIONS

We have proposed the generalized methodology for answer identification with small sized Datasets. The methodology is the combination of extractive feature generation and use of sentence embeddings. Extractive features are obtained with the help of dependency graph that considers inherent grammatical relationships. Lexical semantic features provide clue for word and phrase level similarity in information retrieval systems. SBERT model is used for identification of sentence level textual similarity. Combined approach using pretrained language tools and sentence embeddings with SBERT model is found fruitful for answer identification in small sized datasets. This methodology describes stepwise procedure followed in reading comprehension. It can be visualised as a Learning Tool to demonstrate the task of reading comprehension at elementary/middle school level.

The methodology can be further extended by enhancing contextual features related to various grammatical constituents, reference identification, and negation handling features.

ACKNOWLEDGMENT

The authors would like to thank Isha Kaskhedikar and Aparna Jadhav, for the evaluation of datasets.

REFERENCES

[1] Reimers, N., Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. arXiv preprint arXiv:1908.10084. https://arxiv.org/abs/1908.10084

[2] Rajpurkar, P., Zhang, J., Lopyrev, K., Liang, P. (2016). Squad: 100,000+ questions for machine comprehension of text. arXiv preprint arXiv:1606.05250. https://arxiv.org/abs/1606.05250

[3] Rajpurkar, P., Jia, R., Liang, P. (2018). Know what you don't know: Unanswerable questions for SQuAD. arXiv preprint arXiv:1806.03822. https://arxiv.org/abs/1806.03822

[4] Tan, C., Wei, F., Yang, N., Du, B., Lv, W., Zhou, M. (2018). S-net: From answer extraction to answer synthesis for machine reading comprehension. In Proceedings of the AAAI conference on artificial intelligence, pp. 5940–5947. https://doi.org/10.1609/aaai.v32i1.12035

[5] Dong, H. (2019). Introduction to BERT and Transformer: Pre-trained self-attention models to leverage unlabeled corpus data.

[6] Hewlett, D., Lacoste, A., Jones, L., Polosukhin, I., Fandrianto, A., Han, J., Kelcey, M., Berthelot, D. (2016). Wikireading: A novel large-scale language understanding task over wikipedia. arXiv preprint arXiv:1608.03542. https://arxiv.org/abs/1608.03542

[7] Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Rush, A.M., et al. (2020). Transformers: State-of-the-art natural language processing. In Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations, pp. 38-45. http://doi.org/10.18653/v1/2020.emnlp-demos.6

[8] Gotmare, P.S., Potey, M.M. (2023). Review of Parameters, Approaches and Challenges in Reading Comprehension Systems. In: Tuba, M., Akashe, S., Joshi, A. (eds) ICT Systems and Sustainability. Lecture Notes in Networks and Systems, vol. 516. Springer, Singapore. https://doi.org/10.1007/978-981-19-5221-0_72

[9] Chen, J., Zhang, R., Guo, J., de Rijke, M., Liu, Y., Fan, Y., Cheng, X. (2023). A unified generative retriever for knowledge-intensive language tasks via prompt learning. arXiv preprint arXiv:2304.14856. https://arxiv.org/abs/2304.14856

[10] Daull, X., Bellot, P., Bruno, E., Martin, V., Murisasco, E. (2023). Complex QA and language models hybrid architectures, Survey. arXiv preprint arXiv:2302.09051. https://arxiv.org/abs/2302.09051

[11] Bar-Haim, R., Dagan, I., Berant, J. (2015). Knowledge-based textual inference via parse-tree transformations. Journal of Artificial Intelligence Research, 54: 1-57. https://doi.org/10.1613/jair.4584

[12] Wang, H., Bansal, M., Gimpel, K., McAllester, D. (2015). Machine comprehension with syntax, frames, and semantics. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, pp. 700-706. https://doi.org/10.3115/v1/p15-2115

[13] Palmer, M., Gildea, D., Kingsbury, P. (2005). The proposition bank: An annotated corpus of semantic roles. Computational linguistics, 31(1): 71-106. https://doi.org/10.1162/0891201053630264

[14] Srikumar, V., Roth, D. (2013). Modeling semantic relations expressed by prepositions. Transactions of the Association for Computational Linguistics, 1: 231-242. https://doi.org/10.1162/tacl_a_00223

[15] Shen, D., Lapata, M. (2007). Using semantic roles to improve question answering. In Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL), pp. 12-21.

[16] He, L., Lewis, M., Zettlemoyer, L. (2015). Question-answer driven semantic role labeling: Using natural language to annotate natural language. In Proceedings of the 2015 conference on empirical methods in natural language processing, pp. 643-653. https://doi.org/10.18653/v1/d15-1076

[17] Pizzato, L.A., Mollá, D. (2008). Indexing on semantic roles for question answering. In Coling 2008: Proceedings of the 2nd Workshop on Information Retrieval for Question Answering, pp. 74-81. https://doi.org/10.3115/1641451.1641461

[18] Zettlemoyer, L.S., Collins, M. (2012). Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. arXiv preprint arXiv:1207.1420. https://arxiv.org/abs/1207.1420

[19] Premasiri, D., Ranasinghe, T., Zaghouani, W., Mitkov, R. (2022). DTW at Qur'an QA 2022: Utilising transfer learning with transformers for question answering in a low-resource domain. arXiv preprint arXiv:2205.06025. http://arxiv.org/abs/2205.06025

[20] Wei, Y., Lei, F., Zhang, Y., Zhao, J., Liu, K. (2023). Multi-view graph representation learning for answering hybrid numerical reasoning question. arXiv preprint arXiv:2305.03458. http://arxiv.org/abs/2305.03458

[21] Bousmaha, K.Z., Hamadouche, K., Gourara, I., Hadrich, L.B. (2022). DZ-OPINION: Algerian dialect opinion analysis model with deep learning techniques. Revue d'Intelligence Artificielle, 36(6): 897-903. https://doi.org/10.18280/ria.360610

[22] Al-Ani, J.A., Fasli, M. (2018). Probabilistic relational supervised topic modelling using word embeddings. In 2018 IEEE International Conference on Big Data (Big Data), pp. 2035-2043. https://doi.org/10.1109/bigdata.2018.8622326

[23] Bahdanau, D., Bosc, T., Jastrzębski, S., Grefenstette, E., Vincent, P., Bengio, Y. (2017). Learning to compute word embeddings on the fly. arXiv preprint arXiv:1706.00286. https://arxiv.org/abs/1706.00286

[24] Wehnert, S., Dureja, S., Kutty, L., Sudhi, V., De Luca, E. W. (2022). Applying BERT embeddings to predict legal textual entailment. The Review of Socionetwork Strategies, 16(1): 197-219. https://doi.org/10.1007/s12626-022-00101-3.

[25] Khadhraoui, M., Bellaaj, H., Ammar, M.B., Hamam, H., Jmaiel, M. (2022). Survey of BERT-base models for scientific text classification: COVID-19 case study. Applied Sciences, 12(6): 2891. https://doi.org/10.3390/app12062891

[26] Prottasha, N.J., Sami, A.A., Kowsher, M., Murad, S.A., Bairagi, A.K., Masud, M., Baz, M. (2022). Transfer learning for sentiment analysis using BERT based supervised fine-tuning. Sensors, 22(11): 4157. https://doi.org/10.3390/s22114157

[27] Liu, W., Yi, J., Hu, Z., Gao, Y. (2022). An improved BERT and syntactic dependency representation model for sentiment analysis. Computational Intelligence and Neuroscience, 2022: 5754151. https://doi.org/10.1155/2022/5754151

[28] Liu, X., Hussain, H., Razouk, H., Kern, R. (2022). Effective use of BERT in graph embeddings for sparse knowledge graph completion. In Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing, pp. 799-802. https://doi.org/10.1145/3477314.3507031

[29] Devlin, J., Chang, M.W., Lee, K., Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805. https://arxiv.org/abs/1810.04805

[30] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I. (2017). Attention is all you need. Advances in Neural Information Processing Systems, 30: 5999-6009. https://doi.org/10.48550/arXiv.1706.03762

[31] Khashabi, D., Khot, T., Sabharwal, A., Roth, D. (2018). Question answering as global reasoning over semantic abstractions. In Proceedings of the AAAI Conference on Artificial Intelligence, pp. 1905–1914. https://doi.org/10.1609/aaai.v32i1.11574

[32] Clark, P., Etzioni, O., Khot, T., Sabharwal, A., Tafjord, O., Turney, P., Khashabi, D. (2016). Combining retrieval, statistics, and inference to answer elementary science questions. In Proceedings of the AAAI Conference on Artificial Intelligence, pp. 2580–2586. https://doi.org/10.1609/aaai.v30i1.10325

[33] Manning, C.D., Surdeanu, M., Bauer, J., Finkel, J.R., Bethard, S., McClosky, D. (2014). The Stanford CoreNLP natural language processing toolkit. In Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations, pp. 55-60.

[34] Khashabi, D., Chaturvedi, S., Roth, M., Upadhyay, S., Roth, D. (2018). Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), New Orleans, Louisiana, pp. 252-262. https://doi.org/10.18653/v1/n18-1023

[35] Cucerzan, S., Agichtein, E. (2005). Factoid Question Answering over Unstructured and Structured Web Content. In TREC, 72: 90.

[36] Lee, H., Chang, A., Peirsman, Y., Chambers, N., Surdeanu, M., Jurafsky, D. (2013). Deterministic coreference resolution based on entity-centric, precision-ranked rules. Computational Linguistics, 39(4): 885-916. https://doi.org/10.1162/COLLa.00152

[37] Richardson, M., Burges, C.J., Renshaw, E. (2013). Mctest: A challenge dataset for the open-domain machine comprehension of text. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pp. 193-203.

[38] Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., Soricut, R. (2019). Albert: A lite bert for self-supervised learning of language representations. arXiv Preprint arXiv:1909.11942. https://arxiv.org/abs/1909.11942