Question Answering System

K Sai Chethan
Department of Computer Science
Amrita School of Engineering
Bangalore,India

V Lokesh
Department of Computer Science
Amrita School of Engineering
Bangalore,India

***Abstract*—Question & Answering (Q&A) systems can have a huge impact on the way information is accessed in today's world. In the domain of computer science, Q&A lies at the intersection of Information Retrieval and Natural Language Processing. It is very tedious for anyone to go through the whole document to get answers for their queries. So, there is a need for Question Answering system to make life easier. Question Answering Models are built to automatically answer the question asked in natural language. The system is closed domain question answering system as we are only focused on a specific domain. The model is designed to identify relevant information from input data and generate a natural language response that accurately answers the question. The system can answer questions based on given context. It will extract the answer phrases from the set of paragraphs and give results. The model uses BERT/Transformer architecture to accurately answer the natural language question.**

***Keywords*—Word2vec Model, SIF Model, BERT, Cosine Similarity**

I. Introduction

A significant NLP challenge and a long-standing achievement in artificial intelligence is question answering. A user can ask a question using natural language in QA systems and receive an instant, succinct response. QA systems are now present in phone conversational interfaces and search engines, and they are capable of responding to little informational snippets. However, for more challenging queries, these typically only go as far as giving a list of snippets that the users must then read through to discover the solution to their problem.

Question answering is a critical NLP problem and a long-standing artificial intelligence milestone. A user can ask a question using natural language in QA systems and receive an instant, succinct response. QA systems are now present in phone conversational interfaces and search engines, and they are capable of responding to little informational snippets. However, for more challenging queries, these typically only go as far as giving a list of snippets that the users must then read through to discover the solution to their problem.

Reading comprehension refers to the capacity to comprehend a text and then respond to inquiries about it. Because reading comprehension involves both a grasp of natural language and familiarity with the outside world, it is challenging for machines to perform.

Kevin Clark et al. proposed ELECTRA in March 2020 [5]. It is a two-transformer text encoder, or generator and discriminator model, that has been previously trained by identifying the precise tokens in the source phrase.

The Stanford Question Answering Dataset () is a reading comprehension dataset made up of questions posed by crowd workers on a selection of Wikipedia articles, with the response to each question being a text segment, or span, from the pertinent reading passage, or the question being unanswerable.

The readings in SQuAD cover a wide range of themes, from musical icons to ethereal ideas, and are culled from excellent Wikipedia pages. A passage is a single paragraph from an article, and it can be any length. Each passage in SQuAD is accompanied by reading comprehension questions. You can get the answers to these questions by reading the passage once more. They are based on its substance. Finally, we have a response to each question or a few.

II. LiteratureSurvey

Since there is an increasing need for the QA System, the system automatically responds to particular questions in a brief and accurate manner when the user queries it to extract data from a structured database or a collection of documents written in natural language [1].

Our QA system has numerous real-world applications, including Google Assistant, Alexa, and Cortana.

Closed area: Questions that fall under a specific area, such as medical, automotive maintenance, or education, are dealt with by this type of QA system. We can derive answers from this that are domain-specific. Only a certain kind of queries are accepted in this context (being accepted entails providing answers to the questions) [2].

Open Domain: This QA system is capable of handling almost any kind of question. It depends on general understanding in this regard [3]. In order to respond to inquiries about a baseball game that was played by the US, the very first QA system, called BASEBALL, was created in 1961. It was QA in a closed domain.

The transformer model was first mentioned in the well-known document "Attention is all you need" that Google delivered in 2017 [4]. The NLP Transformer model is an architecture that places a focus on solving problems from sequence to sequence while maintaining extended-span control.

In October 2018, Jacob Devlin and colleagues introduced BERT (Bidirectional Representations from Transformers).

One of the most important ideas that revolutionised NLP methodology was the introduction of pre-trained deep bidirectional models.

Alec Radford et al. introduced GPT- 3, the third-generation autoregressive language model in the GPT, in June 2020. The model has 175B Parameters and is 100 times larger than GPT-2.

III. DATASET

The data sets we have used was created by us. The dataset is in the form of SQuAD dataset. But we have prepared our own dataset in the form of SQuAD dataset.

answer_id	document	question_i	text	answer_st	answer_er	answer_ca	question	is_answer_	context
742637	1379220	680807	there were	0	586		how many	1	there were 28 states and 8
742639	1379220	680809	As for the	586	656		who is the	1	, Odisha, Punjab, Rajasthan,
742642	1379221	680812	Capital: Th	116	275		what is the	1	I can provide you with the n
742644	1379221	680814	Famous da	275	482		what is the	1	I can provide you with the n
742645	1379221	680815	Famous fo	483	617		what is the	1	luru. It is the largest city in t
742651	1379221	680818	GDP: Acco	617	853		what is the	1	f Karnataka is Yakshagana, a
742653	1379221	680822	Total popu	853	959		what is the	1	ath, Mysore masala dosa, M
742656	1379221	680825	the Chief M	1020	1073		who is the	1	e Domestic Product (GSDP)
742657	1379221	680826	Number of	1141	1245		total distri	1	46 billion).Total population:
742659	1379221	680828	Total COVI	1245	1555		total covid	1	as 61,130,704.Chief Ministe
742753	1379222	680922	I can provi	0	325		what is the	1	I can provide you with the n
743196	1379223	681364	According	782	1008		what is the	1	pular dance form in Kerala is
742785	1379222	680954	One of the	325	530		what is the	1	e most up-to-date informat

Fig.1.Dataset

Columns in our dataset:

- Question
- Context
- Answer
- Answer_Start
- Answer_End

IV. Previous Work

Early Q&A models mainly relied on Information Retrieval (Yang et al.), which searches a structured database for information to extract the most appropriate answer to the user's questions, and Rule Based approaches (Madabushi et al.), which use grammatical semantics to categorise questions based on predefined patterns and answer types. As the amount of unstructured data grew over the past ten years, the paradigm moved in favour of statistical models (Ishwari et al.). Information retrieval and rule-based models employ structured queries, whereas statistical techniques can process natural-language inquiries without the need for them. Bayesian classification, maximum entropy models utilising N-gram and Bag of Words based features, and support vector machines (Moschitti et al.), are some examples.

Statistical models perform better than rule-based techniques, however employing them for MRC is insufficient due to their inability to comprehend contextual data. Because of this, statistical models are challenging to scale, especially with the growth in data amount (Cohn et al.). Improved deep learning-based models
with their capacity to self-identify and learn linguistic aspects, based statistical models. As long as there is adequate training data, this makes them very scalable (Ishwari et al.).

Various methods have recently been developed and proposed for enhancing MRC based on deep learning. Due to their low capacity for self-understanding contextual information, traditional word embedding based models like Word2Vec (Mikolov et al.) that use neural networks to learn word associations have proved ineffective for MRC. Word2Vec was outperformed by sentence-based embeddings like SIF (Arora et al.), which could inherit features from the underlying word embeddings.

When using deep contextualised embedding-based language models, such as Embeddings from Language Models (ELMo), each token is represented as a function of the complete input sentence, breaking the tradition of word embeddings. This can get around the drawbacks of earlier embedding-based models, which typically model each word as an average of all of its contexts. in Neumann et al.

Numerous NLP tasks have proved to benefit from language model pre-training (Zhang et al.). A deep contextualised embedding based language model called Bidirectional Encoder Representations from Transformers (BERT) by Google (Devlin et al.) has received a lot of attention lately. The state-of-the-art in many NLP tasks has considerably improved with the release of BERT, especially MRC-based Q&A.

V. Our Approach

We start with embedding-based models to establish our baseline before moving on to MRC-based Q&A. We examine Word2Vec embeddings and SIF embeddings as two methods for doing this. Deep learning models that use embedding look for vector representations of words and sentences. This is done to determine how closely the information in the corpus resembles the inquiry and to obtain the data that most closely matches the query.

We experiment with embedding models first, and then move on to more sophisticated attention-based neural networks. models. BERT is a deep contextualised embedding-based language model that Google Research has created. We begin by selecting the BERT model to test its capacity to efficiently respond to the queries in our dataset. We experiment with settings and fine-tuning after starting with the default BERT-base-uncased. After that, we attempt to combine models to address the problems we encountered when utilising individual models. The performance of our models is then evaluated through comparison and the definition of specific measures.

VI. Methodology

First, we prepare the data then followed by training andvalidation of the model and finally testing the data for theevaluationmetrics.

A. Data Pre-Processing

Datapoints that contain NAN values in the Context or answer or question columns are dropped from the dataframe. We then read the context and extract the text from that file. Newlines (“\n”) in ArticleText are replaced with (“. ”) fullstops. We removed stop words and converted all letters into lower case letters.

- Tokenization: Tokenization is the process of splitting text into individual words or tokens. It breaks down a sentence or paragraph into smaller units, such as words or subwords, which serve as the basic building blocks for further analysis.

- Text Lowercasing: Converting all text to lowercase can help standardize the text and reduce the dimensionality of the data. It ensures that words with different capitalizations are treated as the same, for example, "apple" and "Apple" will both become "apple"
- Stop Word Removal: Stop words are commonly occurring words in a language (e.g., "and", "the", "is") that do not carry significant meaning and can be safely removed from the text. Removing stop words can reduce noise and decrease the dimensionality of the data.
- Punctuation Removal: Removing punctuation marks, such as periods, commas, and quotation marks, helps simplify the text and removes unnecessary noise that may not contribute much to the analysis or modeling process.

B. Approach-1

The meaning of words can be deduced from their contexts using embedding models, which are based on co-occurrence. As our basic models, we take advantage of these models' capabilities and adapt them to a question-answering scenario. Here is a detailed description of how we implemented embedding-based methods:

1. Word2Vec based question answering model:

A neural network is used by Word2Vec to extract word associations from a sizable text corpus. Once trained, this model can predict synonyms, identify similarities within words, and calculate the cosine similarity between different words. The machine can comprehend the semantics of the language since it can quantify word similarity. You can use this to create a Q&A system. On our dataset, we were able to apply the following word2vec embeddings to a question-answering scenario:

1. First, the input data is converted into a list of lists using basic python data manipulation. This data is fed into the word2vec model. The model is then trained for 50 epochs. The embedding size is kept fixed at 100, with a context window of size 8.

2. After the model has been trained, we start providing answers. Our inquiry is broken down into its individual words, and they are then sent to the Word2Vec model. We add up and then average the generated embeddings. This provides the query with an embedding. The response is then expected to be derived from the matching article text, which we separate into individual sentences. Then, to locate embeddings for each sentence in our article content, we employ a similar strategy.

3. Once we obtain the embeddings for the question and each sentence in the response text, we can compare the embeddings of the question and each sentence in the article by using cosine similarity. The model's anticipated output for the given question is the sentence that most closely resembles the query.

This very simple question answering system is able to predict the correct answer with a fair amount of accuracy.

2. SIF based question answering model:

Sentence embeddings include a complete sentence in a vector space, similar to normal word embeddings like Word2Vec. This model can identify similarities between texts after being trained. This can be used to gauge the cosine similarity between various sentences as well as anticipate sentences with similar meanings. You can use this to create a Q&A system. On our dataset, we were able to apply the following SIF embeddings to a Q&A scenario:

1. The glove-wiki-gigaword-100 dataset was used to pre-train a SIF embedding model, which we initialise first. Because the context articles in our dataset are based on wikipedia articles, this dataset serves as our pre-trained base. This also aids in the improvement of our base embeddings because the model learning was constrained by training it only on our 4,000 datapoint dataset.

2. The glove pre-trained SIF embedding model is then trained using all the article text from our own dataset.

3. After the model has been trained, we start providing answers. The SIF embedding for the question is first discovered. The response is then expected to be derived from the matching article text, which we separate into individual sentences. Then, for each sentence in our article text, we locate the SIF embedding.

4. Once we obtain the embeddings for the question and each sentence in the response text, we can compare the embeddings of the question and each sentence in the article by using cosine similarity. The model's anticipated output for the given question is the sentence that most closely resembles the query.

This question answering system is able to predict the correct answer with slightly improved accuracy over Word2Vec embedding model.

C. Approach-2

Transformers have completely revolutionised NLP, particularly in the area of Q&A systems. The concepts presented in that study were first proposed in the work "Attention is all you need" (Vaswani et al.), and the most recent deep learning models have progressively used them to create outstanding outcomes in a variety of NLP tasks. Another transformer that is now highly popular is Google's BERT. This is mostly because a BERT model that has already been trained on a language modelling task can be modified via transfer learning to provide cutting-edge models for a range of applications. A multi-layer bidirectional transformer encoder is used by BERT. . BERT models have been pre-trained on two tasks, namely masked language modeling and next sentence prediction. We start by using the BERTforQuestionAnswering model implementation provided by OpenAI for our Q&A system.

First the input question and paragraph tokens are represented as a single packed sequence as follows: Input Format: [CLS] QUESTION [SEP] CONTEXT [SEP].

BERT generates a sequence of contextualised token

representations from a token sequence (Vaswani et al.). Then, BERTforQuestionAnswering introduces two parameters: a start vector S and an end vector E, for the fine tuning. The dot product between the output of the final hidden layer and the start vector S is used by the model to determine the likelihood that word i from the article context will begin the answer span. The possibility of each word being the conclusion of the answer span is determined in the same way. Finding the log-likelihood of the ideal start and finish positions is the training goal.

BERTforQuestionAnswering predicts the start and end positions in the following decreasing preference order:
1. Short Precise Answer
2. Long Answer
3. [CLS] token is returned when no answer is found in the provided context

The standard BERT-base-uncased pretrained model with BERTforQuestionAnswering is used as our starting point. When compared to embedding-based models, the results are enhanced. After that, we continue to adjust the model.

With our dataset, the BERT model cannot be applied directly. BERT can accept up to 512 tokens, but Our dataset contains answer texts that are substantially longer than 512 (between 3,000 and 12,000 tokens). We resolve this issue by dividing the response sentences into more manageable portions. There are 512 tokens each chunk, and is available for our BERT model to use. We ran into a difficulty with this strategy. Some answers were getting split between two different chunks since we employed a set chunk size. The responses as a result were incorrect. Then, we use the stride strategy proposed by (Devlin et al.), which divides the context into chunks of 512 tokens with a stride of 256 tokens.

We run the BERTforQuestionAnswering model iteratively on each of our chunks to uncover probable responses. It offers a beginning and ending index (span) for the answer if it believes it is in the text; otherwise, it outputs the [CLS] tag. The responses are then combined into a list from the various parts. Now that we have a variety of options, we must select the best one. To determine the answer based on highest similarity to the question text, we experimented with utilising several measures of similarity between the question and the answer text for each chuck. We suggest the following strategy to do this.

BERT+SIF: For this, we employ the SIF model that we trained earlier. The SIF model receives the question and the best response from each chuck, and outputs a similarity score for each response. The final output of the question-answering system is picked to be the one that has the most similarity.

The model we provide is a BERT+SIF model overall. Large answer texts are no problem for it, and it accurately prints the right response.

VII. Cosine Similarity

Here we will be representing the cosine similarity for the 3 models we have used.

Because oour dataset has only 220 question and answer pairs, we could not repeat this process for a large number of pairs. Hence, 150 samples were chosen at random. The cosine similarities obtained were summed for these 150 samples, and then averaged.

Function we used to compare models is:

```
def compare(indexes):
  wv_sim = []
  sif_sim = []
  bert_sim = []
  for i in indexes:
    question = df.iloc[i]['question']
    context = df.iloc[i]['context']
    answer = df.iloc[i]['text']

    if ('yes' in answer.lower()) or ('no' in answer.lower()):
      continue

    ans_sif = model_sif.infer([((answer).split(), 0)])

    w = metrics.pairwise.cosine_similarity(model_sif.infer([((get_answer_cosine(question, contextTosents(context))).split(), 0)]), ans_sif)
    s = metrics.pairwise.cosine_similarity(model_sif.infer([((getAnswer(question, context).iloc[0][0]).split(), 0)]), ans_sif)

    b_ans = getAnswerbert(question, context)
    b = metrics.pairwise.cosine_similarity(model_sif.infer([((b_ans[b_ans.keys()[0]]).split(), 0)]), ans_sif)

    wv_sim.append(w)
    sif_sim.append(s)
    bert_sim.append(b)

  print('WV:', wv_sim)
  print('SIF:', sif_sim)
  print('BERT:', bert_sim)

  return wv_sim, sif_sim, bert_sim
```

```
print('Word2Vec Avg Similarity: ', sum(output[0])/len(output[0]))
print('SIF Avg Similarity: ', sum(output[1])/len(output[1]))
print('BERTwSIF Avg Similarity: ', sum(output[2])/len(output[2]))
```

Output;

Model	**Average Cosine Similarity Over SIF Embeddings** *avg(cosine(sif(actual answer), sif(model answer)))*
Word2Vec	0.5839711
SIF	0.6468621
BERT with SIF	0.8129133

These data unambiguously demonstrate that our BERT with SIF model outperforms the SIF model, the Word2Vec model, and the MRC Q&A job at hand. This contrast enables us to understand why attention-based models are swiftly spreading throughout the field of natural language processing.

VIII. Results

A comparison of results between the three models - Word2Vec, SIF, and BERT+SIF is as follows:

Question-1

```
visualise(1)

WARNING:fse.models.base_s2v:found 1 empty sentences
Question:  who is the prime minister of india
Question:  who is the prime minister of india
Real:  as for the current prime minister of india his name is narendra modi
WV:  as for the current prime minister of india his name is narendra modi
SIF:  as for the current prime minister of india his name is narendra modi
BERT:  narendra modi
```

Question-2

```
visualise(2)

Question:  what is the capital of karnataka
Question:  what is the capital of karnataka
Real:  capital the capital of karnataka is bangalore also known as bengaluru it is the largest city in the state and the center of the states tec
WV:  i can provide you with the most uptodate information available to me as of my knowledge cutoff of september 2021 capital the capital of karr
SIF:  it is the largest city in the state and the center of the states technology industry
BERT:  bangalore
```

Question-3

```
visualise(192)

WARNING:fse.models.base_s2v:found 1 empty sentences
Question:  total covid cases of nagaland
Question:  total covid cases of nagaland
Real:  as for the covid19 situation as of march 6 2023 there have been a total of 43788 cases and 885 deaths in nagaland
WV:  as for the covid19 situation as of march 6 2023 there have been a total of 43788 cases and 885 deaths in nagaland
SIF:  as for the covid19 situation as of march 6 2023 there have been a total of 43788 cases and 885 deaths in nagaland
BERT:  43788
```

Now we will give a general context and ask a question in that

Context: Ashoka University is a pioneer in its focus on providing a liberal education at par with the best in the world. The aim at Ashoka is to help students become well-rounded individuals who can think critically about issues from multiple perspectives, communicate effectively and become leaders with a commitment to public service. An Ashoka education carries a strong emphasis on foundational knowledge, thorough academic research based on rigorous pedagogy, and hands-on experience with real-world challenging. The 2000-plus students on campus, drawn from 30 states and over 243 cities in India and 27 other countries, receive a world-class interdisciplinary education through undergraduate and post-graduate programmes led by internationally renowned faculty.

Question-1: What do they teach at University?

Question-2: What is the name of the University?

Predicted answer:

```
Question:  what do they teach?
0    foundational knowledge , thorough academic research
Name: 0, dtype: object
```

IX. Conclusion

The Word2Vec approach provided a helpful starting point. Given that it is a fairly straightforward method based on sentence similarity and wasn't really created for question-answering tasks, we anticipated that it wouldn't provide a high level of accuracy. Word2Vec cannot be utilised to produce natural language answers since it cannot interpret and understand the material. Its fundamental flaw is that it cannot provide direct answers in a human-like manner; it can only output sentences from the answer text. Even on a challenging dataset like ours, the model did fairly well, earning roughly 58% on our custom criterion. In many cases, the answers it created based on the question's similarity provided the necessary information, and generally, this method produced satisfying results.

The Word2Vec model is not quite as sophisticated as the SIF model. The embeddings are now sentence-based rather than word-based, which is the major factor. Once more, SIF is limited in that it can only guess which line might include the solution while being unable to pinpoint it. It outperformed the Word2Vec model, as expected, achieving roughly 64% on our unique score for our dataset. In fact, we achieved close to 70% on our own metric if we took into account the top 3 options produced by the SIF model.

As anticipated, the Word2Vec and SIF models performed worse than our final BERT+SIF model. The pretraining on the SQuAD dataset and subsequent adaptation to our own dataset resulted in a significant improvement in accuracy. By enhancing the final SIF layer to the BERT model, the model could now provide precise responses to the questions based on an answer text of any length. We were able to reach an 81% score on our unique metric.

In conclusion, depending on the input type, output type, and permissible complexity, there are various approaches to construct a question-answering system. Through this research, we were able to learn more about machine learning and information retrieval and gain first-hand experience with a number of cutting-edge methods for solving this issue. We now feel comfortable testing out some of the newest ensemble models and even considering how we might use the information we've just learned to improve their performance.

X. References

1). Rajpurkar et al. SQuAD2.0. (n.d.). Retrieved December 06, 2020, from https://rajpurkar.github.io/SQuAD-explorer/

2). Yang et al., 2015, M.-C. Yang, D.-G. Lee, S.-Y. Park, H.-C. Rim. Knowledge-based question answering using the semantic embedding space, Expert Syst. Appl., 42 (23) (2015), pp. 9086-9104, 10.1016/j.eswa.2015.07.009

3). Madabushi et al. High Accuracy Rule-based Question Classification using Question Syntax and Semantics, Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers

4). S. Acharya, K. Sornalakshmi, B. Paul and A. Singh, "Question Answering System using NLP and BERT," 2022 3rd International Conference on Smart Electronics and Communication (ICOSEC), Trichy, India, 2022, pp. 925-929, doi: 10.1109/ICOSEC54921.2022.9952050.

5). M. Vinodkumar Sadhuram and A. Soni, "Natural Language Processing based New Approach to Design Factoid Question Answering System," 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 2020, pp. 276-281, doi: 10.1109/ICIRCA48905.2020.9182972.

6). B. K. Jha, C. M. V. Srinivas Akana and R. Anand, "Question Answering System with Indic multilingual-BERT," 2021 5th International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 2021, pp. 1631-1638, doi: 10.1109/ICCMC51019.2021.9418387.