OPEN-DOMAIN
QUESTION & ANSWER

Tahseen Begum, E. Pravallika, N. Sowgna, P. Keerthana

CSE Department, K L University

R.V.S Nagar, Moinabad-Chilkur Rd, near AP Police Academy, Aziz Nagar, Telangana 500075

2010030168@klh.edu.in, 2010030046@klh.edu.in, 2010030344@klh.edu.in, 2010030445@klh.edu.in

***Abstract* – Open-domain question answering (QA) is the task of identifying answers to natural questions from a large corpus of documents. The typical open-domain QA system starts with information retrieval to select a subset of documents from the corpus, which are then processed by a machine reader to select the answer spans.**

Aiming to answer an open domain question based on the knowledge base, we suggest a TANDA algorithm that can automatically extract an adverbial pearl from a simple question and translate it into a KB query. similarity preferences are used to exclude a candidate's start after an easy way to link business. Our method obtained an F1 score of 82.47% in test data. In addition, there is also a series of full bug testing and analysis that can identify features and disabilities of a new dataset.

***Keywords* – Domain question, BERT, SQuAD, QASET, MRC, RC.**

1 Introduction

Questionnaire (QA) is a field of computer science within the fields of information retrieval and natural language processing (NLP), which deals with building programs that automatically answer questions asked by people in the native language. To achieve this, we will use the Python Library and TensorFlow wrapper which enables in-depth learning and AI.

Intends to answer the question in the form of a natural language based on large informal texts. The purpose of the QA is to produce concise answers to summarized questions asked in the original language. This type of retrieval is required with the growth of digital knowledge. Previously QAS was designed for a specific domain and has limited functionality.

Introduce QAS Target to the types of questions most frequently asked by users, the features of the data source, and the correct answer generated. We aim to build a QA web-scale system. Many QA systems prior to issuing an answer perform quiz settings to predict the type of question response.

One significant concern with this approach is that the lexical overlap will make sentence selection easier for the QASENT dataset and might inflate the performance of existing systems in additional natural settings. Our WIKIQA dataset differs from the present QASENT dataset in both question and candidate answer sentence distributions. Questions in QASENT were originally employed in TREC 8-13 QA tracks and were a combination of questions from query logs (e.g., Excite and Encarta) and from human editors.

The questions may well be outdated and do not reflect actuality information needed of a QA system user. against this, questions in WIKIQA were sampled from real queries of Bing without editorial revision. On the sentence side, the candidate sentences in QASENT were selected from documents returned by past participating teams in the TREC QA tracks, and sentences were only included if they shared content words from the questions.

These procedures make the distribution of the candidate sentence skewed and unnatural. compared, 20.3% of the answers in the WIKIQA dataset share no content words with questions. Candidate sentences in WIKIQA were chosen from relevant Wikipedia pages directly, which may be closer to the input of a solution sentence selection module of a QA system.

Open-domain Question Answering (OpenQA) is a vital task in tongue Processing (NLP), which aims to answer an issue within the type of tongue supported large-scale unstructured documents. Recently, there has been a surge in the amount of research literature on OpenQA, particularly on techniques that integrate with neural Machine Reading Comprehension (MRC).

While these research works have the need performance to new heights on benchmark datasets, they need to be rarely covered in existing surveys on QA systems. during this work, we review the most recent research trends in OpenQA, with particular attention to systems that incorporate neural MRC techniques. Specifically, we start with revisiting the origin and development of OpenQA systems.

We then introduce modern OpenQA architecture named "Retriever-Reader" and analyzed the varied systems that follow this architecture yet because of the specific techniques adopted in each of the components. We then discuss key challenges to developing OpenQA systems and offer an analysis of benchmarks that are commonly used. We hope our work would enable researchers to be told of the recent advancement and also the open challenges in OpenQA research, so on stimulate further progress in this field.

The “open-domain” part refers to the lack of the relevant context for any arbitrarily asked factual question. In the above case, the model only takes as the input the question but no article about “why Einstein didn’t win a Nobel Prize for the theory of relativity” is provided, where the term “the law of the photoelectric effect” is likely mentioned. In the case when both the question and the context are provided, the task is known as Reading comprehension (RC).

2 Related Work

2.1 Literature Survey

Table 2.1: Literature Survey

S.no	Authors	Title	Publishing	Techniques & dataset	Pros	Cons
1.	Xunlin Zhan, Yinya Huang, Xiao Dong, Qingxing Caoan , Xiaodan Liang	Explainable reasoning paths for commonsense question answering	Received 16 May 2021, Revised 13 October 2021, Accepted 16 October 2021, Available	A reasonable and explainable framework is proposed to explicitly incorporate external reasoning paths with	A path finder and a hierarchical path learner. To answer a commonsense question, the path	when dealing with the out-of-scope reasoning target, and are

			online 29 October 2021	structured information to explain and facilitate commonsense QA.	finder first retrieves explainable reasoning paths from a large-scale knowledge graph, then the path learner encodes the paths with hierarchical encoders and uses the path features to predict the answers.	unaware of explainable structured information
2.	Zechen Guo	Research and Implementation of Open Domain Question Answering System Based on DuReader Dataset and BIDAF Model	2021 doi:10.1088/1742-6596/1769/1/012033	This article embeds deep learning technology into the system and uses intelligent chat to show them.	An open domain question answering system aims at returning an answer in response to the user's question. The returned answer is in the form of short texts rather than a list of relevant documents.	when dealing with the out-of-scope reasoning target, and are unaware of explainable structured information
3.	Sharon Levy	Open-Domain Question-Answering for COVID-19 and Other Emergent Domains	[v1] Wed, 13 Oct 2021 18:06:14 UTC (6,650 KB)	we incorporate effective re-ranking and question-answering techniques, such as document diversity and multiple answer spans. Our open-domain question-answering system can further act as a model for the quick development of similar systems that can be adapted and modified for other developing emergent domains.	which we can use to efficiently find answers to free-text questions from a large set of documents.	when dealing with the out-of-scope reasoning target, and are unaware of explainable structured information

4.	Sewon Min, Danqi Chen, Luke Zettlemoyer, Hannaneh Hajishirzi	Knowledge guided text retrieval and reading for open domain question answering	arXiv preprint arXiv:1911.03868, 2019	Qualitative analysis to illustrate which components contribute the most to the overall system performance. outperforms competitive baselines on three opendomain QA datasets, WEBQUESTIONS, NATURAL QUESTIONS and TRIVIAQA.	We proposed a general approach for text-based open-domain question answering that integrates graph structure at every stage to construct, retrieve and read a graph of passages. Our retrieval method leverages both text corpus and a knowledge base to find a relevant set of passages and their relations.	when dealing with the out-of-scope reasoning target, and are unaware of explainable structured information

3 Proposed Work

3.1 Model & Techniques

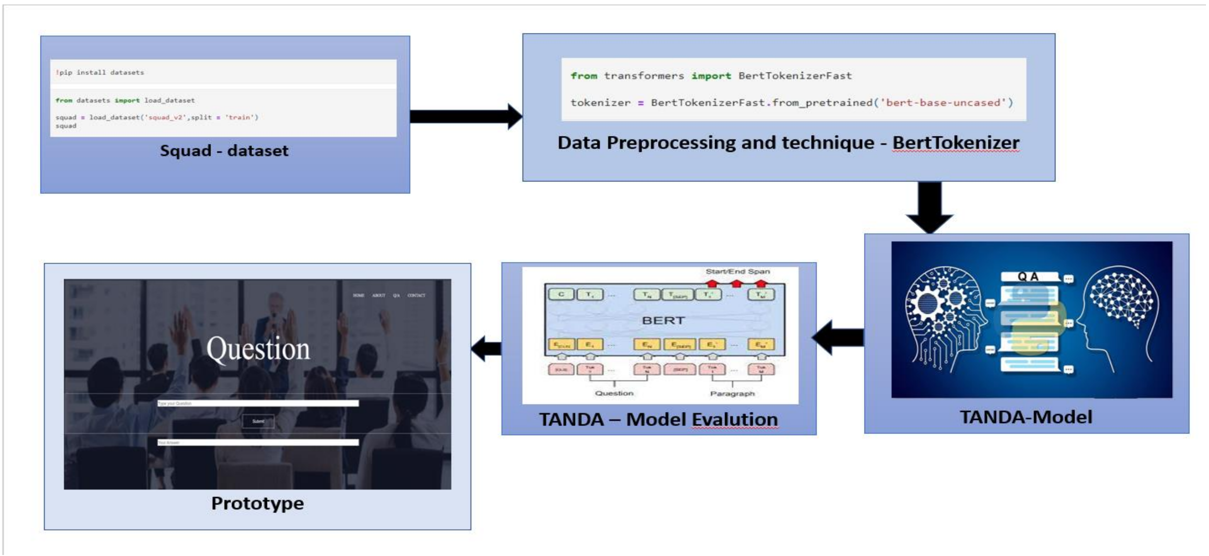
The following are the Model & Techniques we tried to implement in our project

3.1.1 WikiQA

WIKIQA is constructed using a more natural process and is more than an order of magnitude larger than the previous dataset. In addition, the WIKIQA dataset also in- includes questions for which there are no correct sentences, enabling researchers to work on answer triggering, a critical component in any QA system.

3.1.2 SQuAD

Stanford Question Answering Dataset (SQuAD) is a reading comprehension dataset, consisting of questions posed by crowdworkers on a set of Wikipedia articles, where the answer to every question is a segment of text, or span, from the corresponding reading passage, or the question might be unanswerable.



```
!pip install datasets

from datasets import load_dataset

squad = load_dataset('squad_v2',split = 'train')
squad
```

```
from transformers import BertTokenizerFast

tokenizer = BertTokenizerFast.from_pretrained('bert-base-uncased')
```

Fig 3.1.2.1: (Squad Model Flow Chart)

3.2 Model Tuning

Tuning is the process of maximizing a model's performance without overfitting or creating too high of a variance. In machine learning, this is accomplished by selecting appropriate “hyperparameters.”

3.2.1 BERT Model

Apart from the “Token Embeddings”, BERT internally also uses “Segment Embeddings” and “Position Embeddings”. Segment embeddings help BERT in differentiating a question from the text. In practice, we use a vector of 0's if embeddings are from sentence 1 else a vector of 1's if embeddings are from sentence

4 Dataset

4.1 Dataset

Table 4.1 Dataset

Name	Characteristics	Publisher	Accuracy
SQuAD(Stanford Question Answering Dataset)	In SQuAD, the correct answers of questions can be any sequence of tokens in the given text. Because the questions and answers are produced by humans through crowdsourcing, it is more diverse than some other question-answering datasets. SQuAD 1.1 contains 107,785 question-answer pairs on 536 articles. SQuAD2.0 (open-domain SQuAD, SQuAD-Open), the latest version, combines the 100,000 questions in SQuAD1.1 with over 50,000 un-answerable questions written adversarial by crowd workers in forms that are similar to the answerable ones.	Rajpurkar date:2019	96 %

5 Implementation

5.1 Code

```
!pip3 install -q ktrain
```

```
# load 20newsgroups datset into an array
from sklearn.datasets import fetch_20newsgroups
remove = ('headers', 'footers', 'quotes')
newsgroups_train = fetch_20newsgroups(subset='train', remove=remove)
newsgroups_test = fetch_20newsgroups(subset='test', remove=remove)
docs = newsgroups_train.data +  newsgroups_test.data
```

```
import ktrain
from ktrain import text
INDEXDIR = '/tmp/myindex'
```

STEP 1: Create a Search Index

```
text.SimpleQA.initialize_index(INDEXDIR)
text.SimpleQA.index_from_list(docs, INDEXDIR, commit_every=len(docs))
```

STEP 2: Create a QA Instance

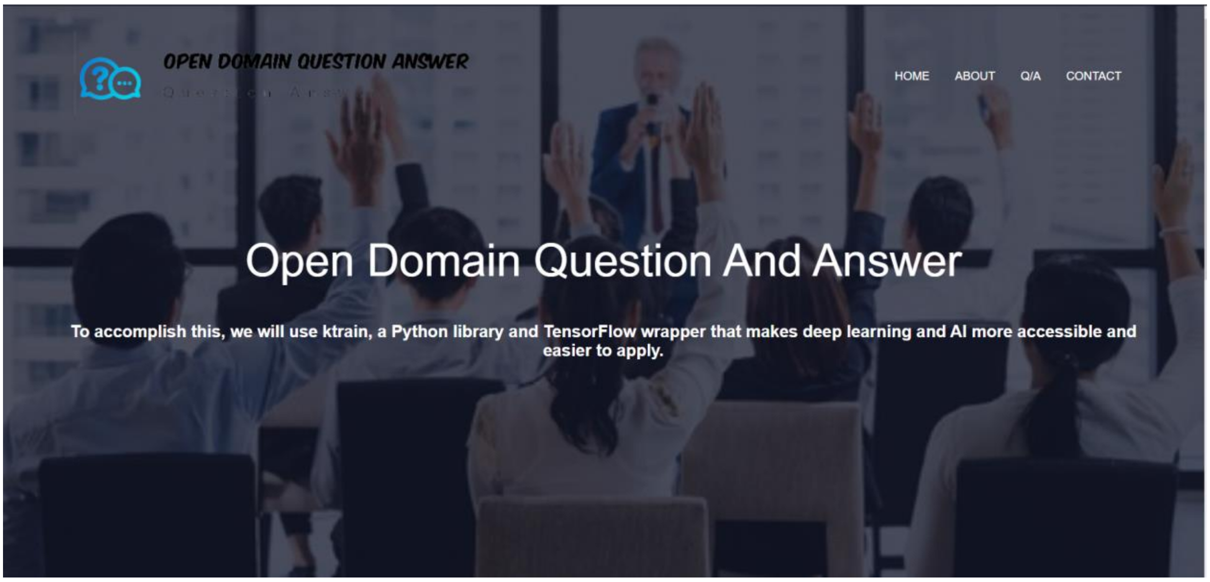
```
qa = text.SimpleQA(INDEXDIR)
```

6 Result

6.1 Tasks Accomplished

- Space Question
- Technical Support Question
- Religious Question

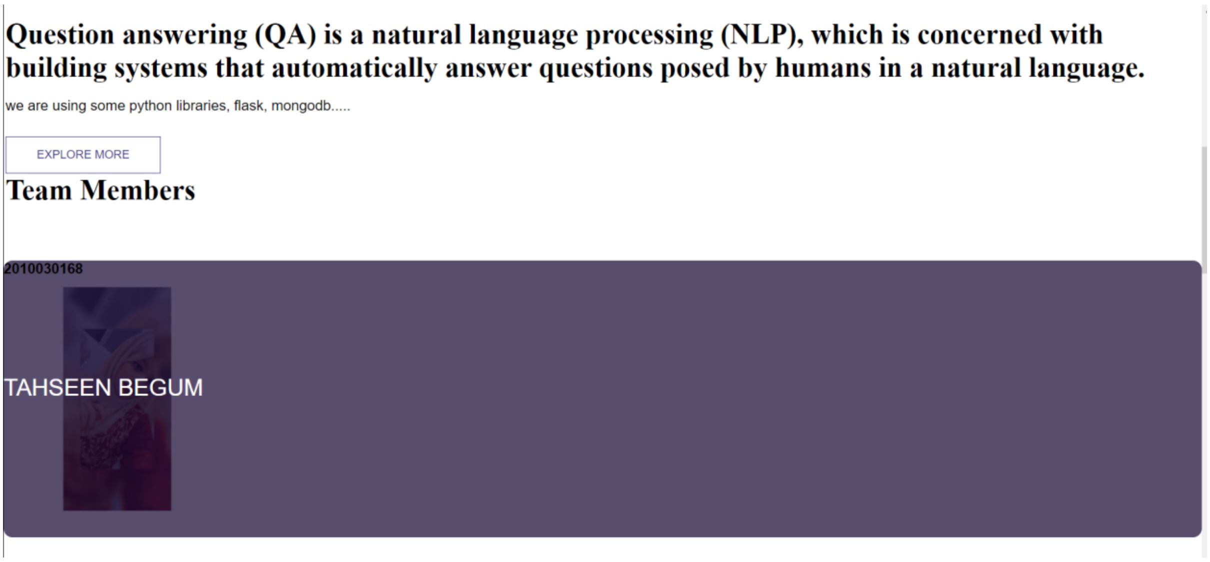
6.2 Outputs



Open Domain Question And Answer

To accomplish this, we will use ktrain, a Python library and TensorFlow wrapper that makes deep learning and AI more accessible and easier to apply.

Fig 6.2.1



Question answering (QA) is a natural language processing (NLP), which is concerned with building systems that automatically answer questions posed by humans in a natural language.

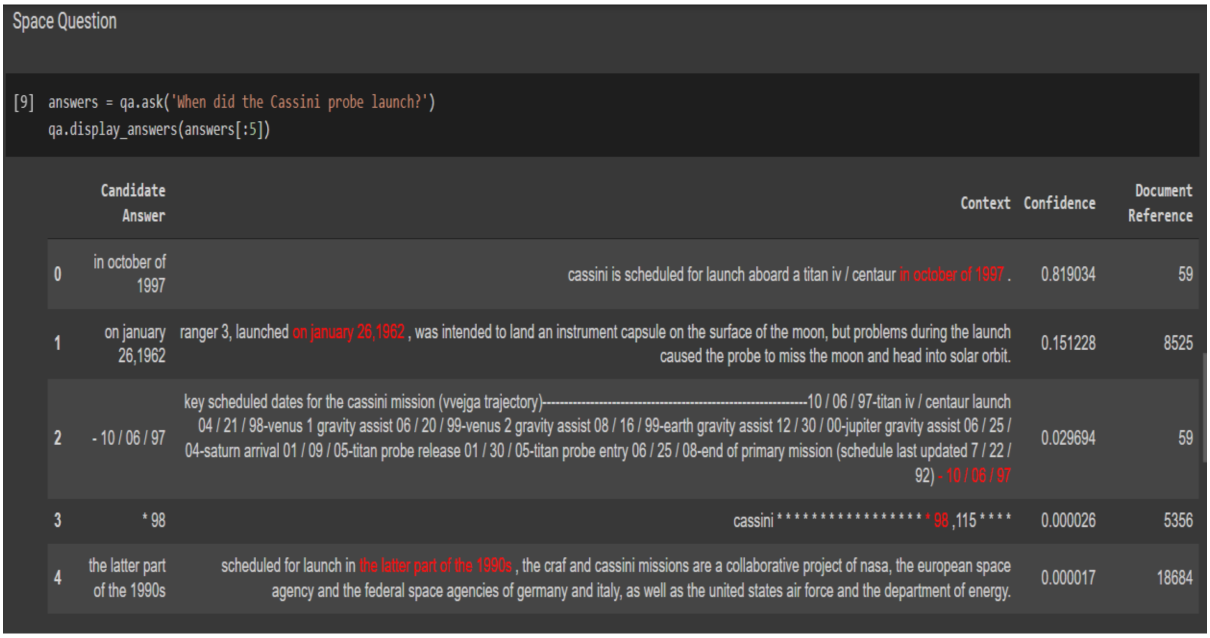
we are using some python libraries, flask, mongodb.....

Team Members

2010030168

TAHSEEN BEGUM

Fig 6.2.2

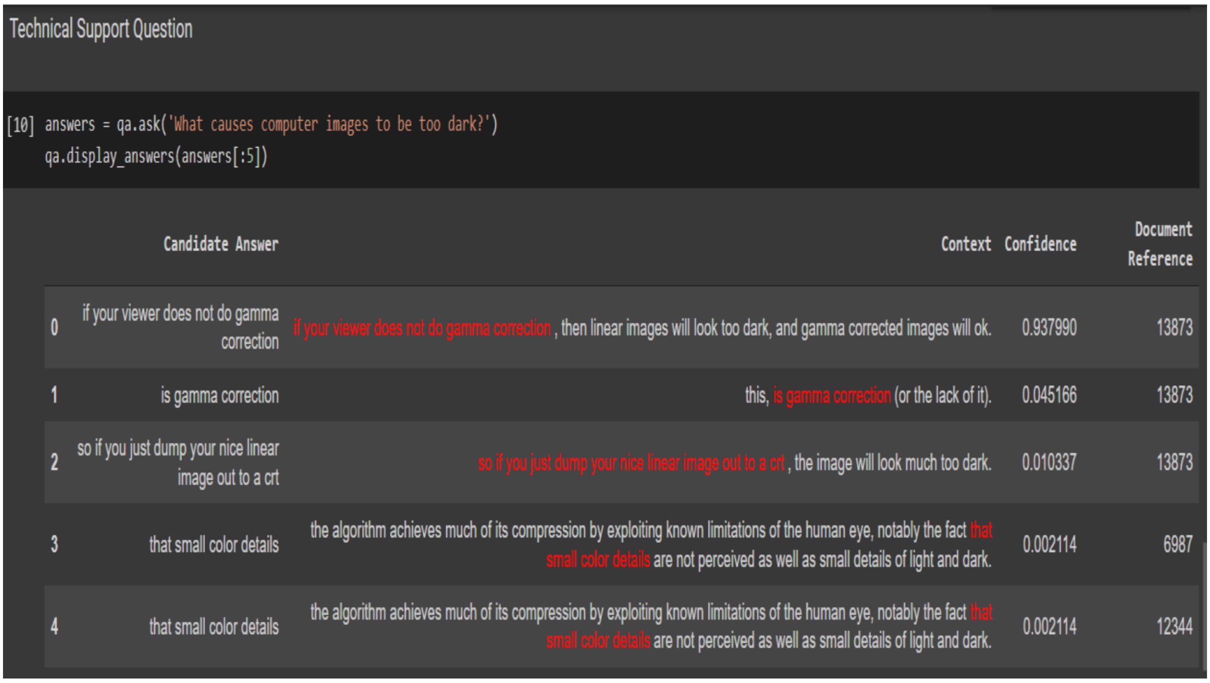


Space Question

```
[9]  answers = qa.ask('When did the Cassini probe launch?')
     qa.display_answers(answers[:5])
```

	Candidate Answer	Context	Confidence	Document Reference
0	in october of 1997	cassini is scheduled for launch aboard a titan iv / centaur in october of 1997 .	0.819034	59
1	on january 26,1962	ranger 3, launched on january 26,1962 , was intended to land an instrument capsule on the surface of the moon, but problems during the launch caused the probe to miss the moon and head into solar orbit.	0.151228	8525
2	- 10 / 06 / 97	key scheduled dates for the cassini mission (vvejga trajectory)--10 / 06 / 97-titan iv / centaur launch 04 / 21 / 98-venus 1 gravity assist 06 / 20 / 99-venus 2 gravity assist 08 / 16 / 99-earth gravity assist 12 / 30 / 00-jupiter gravity assist 06 / 25 / 04-saturn arrival 01 / 09 / 05-titan probe release 01 / 30 / 05-titan probe entry 06 / 25 / 08-end of primary mission (schedule last updated 7 / 22 / 92) - 10 / 06 / 97	0.029694	59
3	* 98	cassini * * * * * * * * * * * * * * * * * * 98 ,115 * * * *	0.000026	5356
4	the latter part of the 1990s	scheduled for launch in the latter part of the 1990s , the craf and cassini missions are a collaborative project of nasa, the european space agency and the federal space agencies of germany and italy, as well as the united states air force and the department of energy.	0.000017	18684

Fig 6.2.3

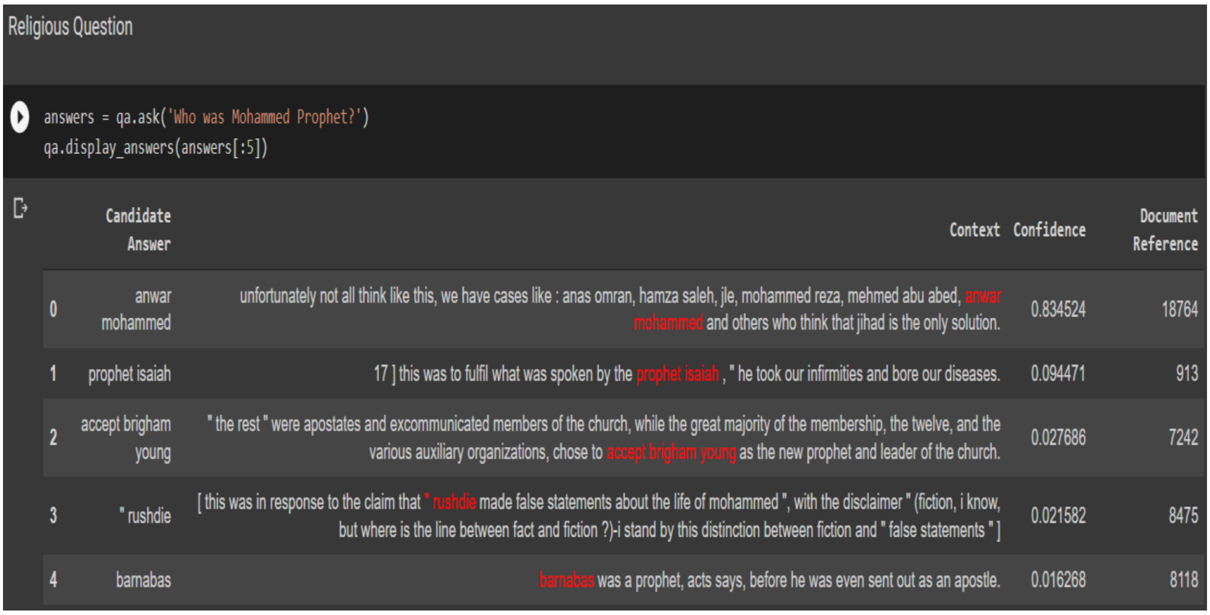


Technical Support Question

```
[10] answers = qa.ask('What causes computer images to be too dark?')
     qa.display_answers(answers[:5])
```

	Candidate Answer	Context	Confidence	Document Reference
0	if your viewer does not do gamma correction	if your viewer does not do gamma correction , then linear images will look too dark, and gamma corrected images will ok.	0.937990	13873
1	is gamma correction	this, is gamma correction (or the lack of it).	0.045166	13873
2	so if you just dump your nice linear image out to a crt	so if you just dump your nice linear image out to a crt , the image will look much too dark.	0.010337	13873
3	that small color details	the algorithm achieves much of its compression by exploiting known limitations of the human eye, notably the fact that small color details are not perceived as well as small details of light and dark.	0.002114	6987
4	that small color details	the algorithm achieves much of its compression by exploiting known limitations of the human eye, notably the fact that small color details are not perceived as well as small details of light and dark.	0.002114	12344

Fig 6.2.4



Religious Question

```
answers = qa.ask('Who was Mohammed Prophet?')
qa.display_answers(answers[:5])
```

	Candidate Answer	Context	Confidence	Document Reference
0	anwar mohammed	unfortunately not all think like this, we have cases like : anas omran, hamza saleh, jle, mohammed reza, mehmed abu abed, anwar mohammed and others who think that jihad is the only solution.	0.834524	18764
1	prophet isaiah	17] this was to fulfil what was spoken by the prophet isaiah , " he took our infirmities and bore our diseases.	0.094471	913
2	accept brigham young	" the rest " were apostates and excommunicated members of the church, while the great majority of the membership, the twelve, and the various auxiliary organizations, chose to accept brigham young as the new prophet and leader of the church.	0.027686	7242
3	" rushdie	[this was in response to the claim that " rushdie made false statements about the life of mohammed ", with the disclaimer " (fiction, i know, but where is the line between fact and fiction ?)-i stand by this distinction between fiction and " false statements "]	0.021582	8475
4	barnabas	barnabas was a prophet, acts says, before he was even sent out as an apostle.	0.016268	8118

Fig 6.2.5

7 Discussion

First, we download the SQuAD v2 dataset. This training process needs us to pass a context as input and add optimize on two integer values, answer_start, and answer_end as labels. We have context and answer_start.

Here, the answer_start marks the character position within the context where we can find the start of our answer. But we also need an answer_end, which does not exit vet. That looks good, however, we will find that some answers do not include answers, like this: The reason we are not returning an answer here is that there is no answer, so we can either ignore this attempt to train our QA model to identify when there is not an answer by setting answer_start and answer_end to a 'no answer values like 1 and 0. we will keep the samples and zero no answers.

8 Conclusion

8.1 Conclusion

To accomplish this, we will use ktrain, a Python library, and a TensorFlow wrapper that makes deep learning and AI more accessible and easier to apply. ktrain is free, and open-source.

1. Uses the search index to locate documents that contain words in the question

2. Extracts paragraphs from these documents for use as contexts and uses a BERT model pretrained on the SQuAD dataset to parse out candidate answers

3. Sorts and prunes candidate answers by confidence scores and returns results

8.2 Future Work

In future work, we plan to improve the Bert modules with extended contexts (i.e. more than one sentence) and add some spatial reasoning. We also want to improve the retrieval by crawling relevant documents from web search engines instead of using snippets. This could be a good method to find more sentences with supported answers.

9 References

[1] Liu, Qian, Bei Chen, Jiaqi Guo, Zeqi Lin, and Jian-guang Lou. "TAPEX: Table pretraining via learning a neural SQL executor." arXiv preprint arXiv:2107.07653 (2021)

[2] Hu, Zhe, Zuohui Fu, Cheng Peng, and Weiwei Wang. "Enhanced Sentence Alignment Network for Efficient Short Text Matching." In Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020), pp. 34-40. 2020.

[3] Soldaini, Luca, and Alessandro Moschitti. "The cascade transformer: an application for efficient answer sentence selection." arXiv preprint arXiv:2005.02534 (2020).

[4] Shi, Qi, Qian Liu, Bei Chen, Yu Zhang, Ting Liu, and Jian-Guang Lou. "LEMON: Language-Based Environment Manipulation via Execution-Guided Pre-training." arXiv preprint arXiv:2201.08081 (2022).

[5] Matt Crane. 2018. Questionable answers in question answering research: Reproducibility and variability of published results. Transactions of the Association for Computational Linguistics, 6:241–252.

[6] Guo, Zechen, Fucheng Wan, and Ning Ma. "Research and Implementation of Open Domain Question Answering System Based on DuReader Dataset and BIDAF Model." In Journal of Physics: Conference Series, vol. 1769, no. 1, p. 012033. IOP Publishing, 2021.

[7] Levy, Sharon, Kevin Mo, Wenhan Xiong, and William Yang Wang. "Open-Domain Question-Answering for COVID-19 and Other Emergent Domains." arXiv preprint arXiv:2110.06962 (2021)