Unstructured Information Processing Project

NLP: Question Answering System

Harpreet Virk & Dhruv Sabharwal

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 deals with building intelligent systems that can provide answers, for user generated queries, in a natural language. From improved search engine results to more personalised conversational chatbots, Q&A systems that can retrieve desired information accurately and quickly, without having to skim through the entire corpus, can serve as valuable assets in industry, academia, and our own personal lives. Two real world scenarios where Q&A systems can find application are Law Firms and HR Departments. In the former some information may be needed (for citation in a current case) from the records of thousands of old cases and using a Q&A system can save huge amounts of time and manpower. In the latter case, an employee may need to look up certain rules, say for vacations, and using a Q&A system can help them by giving desired answers to questions of the form “How many vacation days am I allowed in a year?”.

Introduction

Machine Reading Comprehension (MRC) is a subfield in Q&A systems that aims to answer questions based on reading and understanding contextual text (*Pan et al.*). Automating the task of understanding both - natural language, along with the contextual information, in order to answer MRC questions is a complex and cumbersome task. We begin this paper by examining the progress in research of Q&A systems, focusing primarily on MRC models. We then discuss our step-wise approach for MRC, and describe our implementation. The results are then compared with state-of-the-art models, followed by a concluding discussion.

Previous Work

Early Q&A models have primarily relied on Information Retrieval, looking for information in a structured database in order to extract the most appropriate answer to the user’s questions (*Yang et al.*), and Rule Based approaches that rely on grammatical semantics to classify questions based on predefined patterns and answer types (*Madabushi et al.*). The paradigm had shifted in the last decade towards statistical models to cater to the growing volume of unstructured data (*Ishwari et al.*). Statistical approaches do not require structured queries which are used by Information Retrieval & Rule Based models, and can process queries in natural language. Support vector machines (*Moschitti et al.*), bayesian classification, and maximum entropy based models using N-gram and Bag of Words based features (*Ittycheriah et al.*) are some techniques used in statistical Q&A models.

Although statistical models outperform rule based approaches, using them for MRC is not sufficient because of their limitation in understanding contextual information. This makes statistical models hard to scale, especially with the increasing volume of data (*Cohn et al.*). Deep learning based models improve upon statistical models with their ability to self-identify and learn linguistic features. This makes them highly scalable, as long as there is enough training data (*Ishwari et al.*).

Recently, various techniques have been proposed and developed for improving deep learning based MRC.

Traditional word embedding based models such as Word2Vec (Mikolov et al.) that use neural networks to learn word associations have been inefficient for MRC due to their limited ability in self-understanding contextual information. Sentence based embeddings such as SIF (Arora et al.) improved upon Word2Vec with their ability to inherit features from the underlying word embeddings.

Deep contextualised embedding based language models such as Embeddings from Language Models (ELMo) breaks the tradition of word embeddings by incorporating sub-word units by representing each token as a function of the entire input sentence. This can overcome the limitations of previous embedding based models where each word is usually modeled as an average of their multiple contexts. (Neumann et al.).

Language model pre-training has shown to be effective for improving many NLP tasks (Zhang et al.). Bidirectional Encoder Representations from Transformers (BERT) by Google (Devlin at al.) is a pre-trained deep contextualised embedding based language model, which has gained a lot of attention recently. The release of BERT has significantly improved the state-of-the-art in a number of NLP tasks, particularly MRC based Q&A.

At the time of writing this, the **state-of-the-art** performance on the three most prominent Q&A datasets is as follows:

- 12 out of the top 20 models on the **SQuAD 2.0** (Rajpurkar et al.) dataset ranking leaderboard are based on BERT. Other prominent models are SA-Net & ELECTRA. ELECTRA is a text encoder model that is trained to distinguish real input tokens from plausible fakes, which is used to learn effective language representations (Clark et al.).

- 14 out of the top 20 models on **NQ** (Kwiatkowski et al.) long-answer dataset ranking leaderboard are based on BERT. Other prominent models are RikiNet. RikiNet contains a dynamic paragraph dual-attention reader and a multi-level cascaded answer predictor that dynamically represents the document and question by utilizing a set of complementary attention mechanisms (Liu et al.).

- 13 out of top 20 models on **CoQA** (Reddy et al.) dataset ranking leaderboard are based on BERT. Other prominent models are TR-MT & XLNet. XLNet improved upon BERT's input mask based pre-training - which causes pretrain finetune discrepancy - by proposing learning of bidirectional contexts by maximizing the expected likelihood over all permutations of the factorisation order instead of input masks based learning (Yang et al.).

Our Approach

We approach MRC based Q&A by beginning with embedding based models to first define our baseline. For this, we look at two approaches - Word2Vec embeddings and SIF embeddings. Embedding based deep-learning models try to find vector representations for words and sentences. This is used to find the similarity between the question and the information present in the corpus, which is then used to retrieve the information that most closely resembles the query.

After experimenting with embedding models, we move on to more advanced attention based neural models. BERT is a pre-trained deep contextualised embedding based language model that has been developed by Google Research. We begin by picking up the BERT model to experiment with its ability to answer the questions in our dataset efficiently. Starting with vanilla *BERT-base-uncased*, we experiment with fine-tuning, and settings. We then try combining models to address the issues we faced while using individual models. We then compare and define custom metrics to understand the performance of our models.

Implementation

Data Input: We use Question-Answer Dataset by Rachael Tatman as our dataset, which has the following attributes:

1. ***ArticleTitle*** is the name of the Wikipedia article from which questions and answers initially came.
2. ***Question*** is the question.
3. ***Answer*** is the answer.
4. ***DifficultyFromQuestioner*** is the prescribed difficulty rating for the question as given to the question-writer.
5. ***DifficultyFromAnswerer*** is a difficulty rating assigned by the individual who evaluated and answered the question, which may differ from the difficulty in field 4.
6. ***ArticleFile*** is the name of the file with the relevant article

The dataset is segmented into three sub-datasets, each corresponding to data collected in a different year. We begin by reading and merging the three subsets into a single dataframe.

Data Pre-processing: Datapoints that contain *NAN* values in the ***ArticleFile*** or ***Answer*** columns are dropped from the dataframe. We then read the ***ArticleFile*** names for each datapoint, and extract the article text from that file into a new column ***ArticleText***. Newlines (*"\n"*) in ***ArticleText*** are replaced with (*". "*) fullstops. The ***DifficultyFromQuestioner, DifficultyFromAnswerer,*** and ***ArticleFile*** are then dropped.

Data Cleaning: The text in ***Question, Answer,*** and ***ArticleText*** columns is cleaned by removing [^\w\s\.\?]. The resultant text is converted into lower-case characters.

Approach 1: Embedding Models

Embedding models are based on co-occurrence, and are able to extract the meaning of words using the contexts in which they appear. We leverage the ability of such models and adapt them to a question answering scenario as our baseline models. Our implementations of embedding based approaches is described in detail as follows:

1. **Word2Vec based question answering model:** Word2Vec uses a neural network to learn word associations from a large text corpus. Once trained, this model is able to detect similarity within words and can be used to predict synonyms, as well as measure the cosine similarity between distinct words. The ability to measure the similarity between words enables the machine to understand the semantics of the language. This can be leveraged to build a Q&A system. We were able to adapt word2vec embeddings to a question answering scenario on our dataset as follows:

    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. Once the model is trained, we begin answering questions. We split our question into its component words, and pass them into the Word2Vec model. The generated embeddings are summed together, and then averaged. This gives us an embedding for the question.

Then we move onto the corresponding article text, from which the answer is supposed to be generated, and split it into individual sentences. We then use a similar approach to find embeddings for each sentence in our article text.

3. Once we have the embeddings for the question, and for every sentence in the answer text, we use cosine similarity to find the similarity between the embeddings of the question and each article sentence. The sentence which has the highest similarity with the question is predicted as the output of the model for the given question.

In this way, the Word2Vec model selects a sentence from the article text that has the highest Word2Vec embedding similarity with given question, and outputs it as the answer. 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:** Similar to regular word embeddings such as Word2Vec, sentence embeddings embed a full sentence into a vector space. Once trained, this model is able to detect similarity within sentences. This can be used to predict sentences with similar meanings, as well as measure the cosine similarity between distinct sentences. This can be leveraged to build a Q&A system. We were able to adapt SIF embeddings to a Q&A scenario on our dataset as follows:

    1. We begin by initialising a SIF embedding model pre-trained using the *glove-wiki-gigaword-100* dataset. This dataset is used as our pre-trained base because it is based on wikipedia articles, which is what the context articles in our dataset are based on. This also helps us improve our base embeddings, since training merely on our 4,000 datapoint dataset limited the model learning.
    2. The glove pre-trained SIF embedding model is then trained using all the article text from our own dataset.
    3. Once the model is trained, we begin answering questions. We first find the SIF embedding for the question. Then we move onto the corresponding article text, from which the answer is supposed to be generated, and split it into individual sentences. We then find the SIF embedding for each sentence in our article text.
    4. Once we have the embeddings for the question, and for every sentence in the answer text, we use cosine similarity to find the similarity between the embeddings of the question and each article sentence. The sentence which has the highest similarity with the question is predicted as the output of the model for the given question.

    In this way, the SIF embeddings help us select a sentence from the article text that has the highest embedding similarity with the given question, and outputs it as the answer. This question answering system is able to predict the correct answer with slightly improved accuracy over Word2Vec embedding model.

Approach 2: Attention Based Neural Models

Transformers have taken the NLP world by storm, especially in the field of Q&A systems. They were first introduced in the paper "Attention is all you need" (Vaswani et al.), and the latest deep learning models have increasingly employed the concepts discussed in that paper to produce impressive results in all sorts of NLP tasks. Google's BERT is another type of transformer which has become very popular today. A major reason for this is because a BERT model pre-trained on a language modelling task can be adapted, using transfer learning, to create state-of-the-art models for a variety of tasks. BERT uses a multi-layer bidirectional transformer encoder. Its self-attention layer performs self-attention in both directions. BERT models have been pre-trained on two tasks, namely masked language modeling and next sentence

prediction. Two major variants of BERT as proposed by Google Research are available for use, BERT base with 12 transformer layers and BERT large with 24 transformer layers (Devlin et al.). 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]

Given a token sequence, BERT outputs a sequence of contextualized token representations (Vaswani et al.). BERTforQuestionAnswering then introduces two parameters for the fine tuning: a start vector S, and an end vector E. The model calculates the likelihood of word i from article context to be the start of the answer span as the dot product between the output of the last hidden layer and the start vector S. The same is done to calculate the likelihood of each word as the end of the answer span. The training objective is to find the log-likelihood of the correct start and end position. (Zhang et al.)

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

We begin with the vanilla BERT-base-uncased pretrained model with BERTforQuestionAnswering. The results obtained are improved, when compared to the embedding based models. We then proceed to fine-tune the model.

BERTforQuestionAnswering requires the training dataset answers as context windows - with start and end positions. We observed that our dataset has variations in answer types. About 30% of the answers were in the form of YES or NO. From the remaining, some answers were context windows that were picked up directly from the text, while some were descriptive answers that were not directly picked from the text (manually written). We were thus unable to use this dataset for fine-tuning our BERTforQuestionAnswering Model. We decided to pre-train our model on the SQuAD question-answer dataset provided by Stanford. The dataset differs only in the length of contextual text provided for answering questions - all less than 512 tokens, but the source of the text remains wikipedia articles like our dataset. The answers in this dataset have a uniform representation (beginning and ending token index) and provide a solid foundation for our Q&A system.

The BERT model cannot directly be used with our dataset. BERT takes in a maximum of 512 tokens, but the answer texts in our dataset are much larger than 512 (in the range of 3,000-12,000 tokens). We accommodate this problem by splitting the answer texts into smaller chunks. Each chunk is 512 tokens, and can be fed into our BERT model. We encountered a problem with this approach. Since we used a fixed chunk size, some answers were getting split between two different chunks. This resulted in incorrect answers. We then apply (Devlin et al.) approach of using strides - splitting the context into chunks of 512 tokens with a stride of 256 tokens. Although this increased the number of chunks that we would need to iteratively check for answers, it reduced the possibility of splitting of answers.

We iteratively run the BERTforQuestionAnswering model on all our chunks, and find potential answers from each. If it feels that the answer to the question is present in the text, it provides its beginning and ending index (span), otherwise it provides the [CLS] tag as output. The answers from the various chunks are then turned into a list. Now we have to choose the best among these several choices. We experimented with using different measures of similarity between the question and the answer text for each chuck to find the answer based on maximum similarity to the question text. We propose a way to achieve this as follows:

BERT+SIF: We use the SIF model we had trained earlier for this purpose. The question and the best possible answer from every chuck is fed into the SIF model, and we get a similarity score for each answer as output. The one which then has the highest similarity is chosen as the final output of the question answering system.

Overall, the model proposed by us is a BERT+SIF model. It is able to handle large answer texts and also outputs the correct answer with a high degree of accuracy.

Results

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

QUESTION	ACTUAL ANSWER	WORD2VEC ANSWER	SIF ANSWER	BERT+SIF ANSWER
do all ducks quack	no	diving ducks and sea ducks forage deep underwater	as aforementioned though very few ducks actually do quack	most ducks other than female mall ards and domestic ducks do not quack
what are four species that are commonly referred to as kangaroos	the red kangaroo the eastern grey kangaroo the western grey kangaroo and the antilopine kangaroo	there are four species that are commonly referred to as kangaroos	there are four species that are commonly referred to as kangaroos	the red kangaroo the anti lo pine kangaroo and the eastern and western grey kangaroo
why did cleveland want to hide his cancer surgery from the public	because of the financial depression of the country	the public treasury	because of the financial depression of the country cleveland decided to have surgery performed on the tumor in secrecy	financial depression of the country cleveland decided to have surgery performed on the tumor in secrecy to avoid further market panic
what established a trading post on the island in 1819	british east india company	the british east india company established a trading post on the island in 1819	the british east india company established a trading post on the island in 1819	british east india company
how do elephants communicate over long distances	by producing and receiving low frequency sound infrasound	elephants have a very long childhood	elephants are observed listening by putting trunks on the ground and carefully moving their very sensitive feet	elephants are observed listening by putting trunks on the ground and carefully moving their very sensitive feet
what did roosevelt do to improve his physical condition	roosevelt took up exercise	roosevelt timeline	roosevelt underwent a physical examination	to combat his poor physical condition his father compelled the young roosevelt to take up exercise
did coolidge graduate from black river academy	yes	coolidge graduated from black river academy vermont but failed his initial	coolidge graduated from black river academy vermont but failed his initial	coolidge graduated from black river academy vermont

		entrance exam to amherst college	entrance exam to amherst college	

In order to compare our results with **state-of-the-art** models, we looked at recent works done in Q&A systems on three prominent datasets - SQuAD (Rajpurkar et al.), Natural Questions (NQ) (Kwiatkowski et al.), and CoQA (Reddy et al.) (we have already talked about the approaches adopted for Q&A on these datasets in the *Previous Work* section of this report). However, we realised that we could not directly compare our results with models built to solve Q&A on these datasets. SQuAD is a MRC dataset and CoQA is a conversational Q&A dataset, both with questions answerable using accompanying short <512 token paragraphs. NQ uses >512 token context texts, but contains answers as direct context windows. Since our dataset contains varying types of answers, we could not directly compare our results with those on these datasets. We could not find existing research on Q&A systems built using a dataset that are similar to ours, which could be used for insightful comparisons. Also, our dataset has no submissions on kaggle showing a high level of accuracy, and hence a comparison could not be made.

Moreover, as our dataset does not have uniform answer types (YES /NO /ContextWindowAnswer /ManualAnswer), we also faced the problem of how we can accurately measure the performance of our model, and what metrics to use for this purpose. Metrics like vanilla accuracy and F1 score could not be directly calculated for our model, since our answers - although correct according to the meaning conveyed, sometimes differed when compared to the expected answer from the dataset.

We thought of a workaround to this problem. We decided to calculate the cosine similarity between the actual answers and the answers outputted by our models. This was repeated for all 3 of our models, in the following way.

1. First, the SIF embedding is calculated for the actual answer string.
2. Next, the SIF embedding is calculated for the output answer string generated by our model.
3. Finally, the cosine similarity was calculated between these 2 SIF embeddings.
4. Because of lack of compute power, we could not repeat this process for a large number of samples (questions). Hence, 200 samples were chosen at random. The cosine similarities obtained were summed for these 200 samples, and then averaged.

This number between 0 and 1 gives us a fair estimate for how well our model was performing. We have provided the numbers below.

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 numbers clearly show that our BERT with SIF model performs the best in the task of MRC Q&A, followed by the SIF model and then finally the Word2Vec model. This comparison helps us appreciate why attention based models are quickly becoming ubiquitous in the domain of natural language processing.

Discussion & Conclusion

Building a robust Question Answering system is a difficult task. The various types of answers (yes/no, descriptive, spans) that a user may expect for a particular question add further complexity to this problem. We started with very simple approaches like Word2Vec and SIF embeddings and then moved onto more advanced attention based state-of-the-art models like BERT.

The Word2Vec model gave us a useful baseline. We expected it to not give a high level of accuracy since it is a very simple approach based on sentence similarity and not actually developed for question answering tasks. Word2Vec does not comprehend and understand the information, and cannot be used to generate answers in natural language. It's main drawback is that it can only output sentences from the answer text, and not give direct answers in a human-like fashion. Nevertheless, the model performed quite well, scoring about 58% on our custom metric, even on a complex dataset like ours. The answers it generated based on the similarity to the question contained required information in many occasions and overall this approach gave us satisfying results.

The SIF model is slightly more advanced than the Word2Vec model. The main reason is that the embeddings are now sentence based and not word based. SIF again faces the drawback of only being able to tell which sentence might contain the answer, but not actually being able to pin-point the answer. We expected it to perform better than the Word2Vec model, which it did, scoring about 64% on our custom metric, on our dataset. If we considered the top 3 choices outputted by the SIF model we actually got close to 70% on our custom metric.

As expected our final BERT+SIF model outperformed both the Word2Vec and SIF model. Pretraining on the SQuAD dataset and then adapting it to our own dataset led to large increase in accuracy. The model was now able to give pin-point answers to the questions based on an answer text of any length, made possible by augmenting the final SIF layer to the BERT model. We were able to touch a score of 81% on our custom metric.

In conclusion, there are several ways of building a question answering system based on the input type, output type and permitted complexity. This project allowed us to dive deeper into this area of machine learning and information retrieval and gave us hands on experience with several modern ways of approaching this problem. We feel we are now confident to try out some of the latest ensemble models and even think of ways to apply our recently gained knowledge to try to make them work better.

References

1. Rajpurkar et al. SQuAD2.0. (n.d.). Retrieved December 06, 2020, from https://rajpurkar.github.io/SQuAD-explorer/
2. Dwivedi et al. 2013, S.K. Dwivedi, V. Singh, Research and reviews in question answering system. Procedia Technology., 10 (2013), pp. 417-424, 10.1016/j.protcy.2013.12.378
3. 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
4. Ishwari et al. Advances in Natural Language Question Answering: A Review. arXiv 2019, arXiv:1904.05276 [cs.CL]
5. 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
6. Ittycheriah A., Franz M, Zhu WJ, Ratnaparkhi A. and Mammone R. J., "IBM's

statistical question answering system," in Proceedings of the Text Retrieval Conference TREC-9, 2000.
7. Moschitti A. "Answer filtering via text categorization in question answering systems," in Proceedings of the 15th IEEE International Conference on Tools with Artificial Intelligence, 2003, pp. 241-248.
8. Neumann et al. Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. arXiv preprint arXiv:1802.05365, 2018.
9. Mikolov et al. Efficient Estimation of Word Representations in Vector Space. 2013, arXiv:1301.3781v3 [cs.CL]
10. Cohn et al. Active Learning with Statistical Models, Journal of Artificial Intelligence Research 4 (1996) 129-145
11. Arora et al. A SIMPLE BUT TOUGH-TO-BEAT BASELINE FOR SENTENCE EMBEDDINGS, ICLR 2017.
12. Kwiatkowski et al. Natural Questions: a Benchmark for Question Answering Research, Transactions of the Association of Computational Linguistics (2019) (to appear).
13. Park et al, Question Answering on the SQuAD Dataset, Retrieved from https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1174/reports/2761899.pdf
14. Devlin et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, arXiv:1810.04805 [cs.CL]
15. Reddy et al. CoQA: A Conversational Question Answering Challenge, TACL (presented at NAACL 2019), arXiv:1808.07042 [cs.CL]
16. Liu et al, RikiNet: Reading Wikipedia Pages for Natural Question Answering. Retrieved from https://www.aclweb.org/anthology/2020.acl-main.604.pdf
17. Clark et al. ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators, ICLR 2020 Conference Blind Submission
18. Yang et al. XLNet: Generalized Autoregressive Pre Training for Language Understanding, arXiv:1906.08237v2 [cs.CL] 2 Jan 2020
19. Viswani et al. Attention Is All You Need, arXiv:1706.03762 [cs.CL]
20. Zhang et al, BERT for Question Answering on SQuAD 2.0, Retrieved from https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1194/reports/default/15848021.pdf
21. Muttenthaler et al. Unsupervised Evaluation for Question Answering with Transformers, arXiv:2010.03222v1 [cs.CL] 7 Oct 2020
22. Pan et al. Frustratingly Easy Natural Question Answering, arXiv:1909.05286v1 [cs.CL] 11 Sep 2019
23. Alberti et al, A BERT Baseline for the Natural Questions, arXiv:1901.08634v3 [cs.CL] 9 Dec 2019
24. Question-Answer Dataset, Rachael Tatman, Retrieved from https://www.kaggle.com/rtatman/questionanswer-datasethttps://web.stanford.edu/class/archive/cs/cs224n/cs224n.1194/reports/default/15848021.pdf