Question Answering

Computational Linguistics I
Pedagogical Project

Ramya Rao Basava
Department of Computer Science
University of British Columbia

Sang-Wha Sien
Department of Computer Science
University of British Columbia

Abstract

Question answering is a field of natural language processing, that involves methods which enable computer systems to understand and answer human questions, in natural language. It is an area which has many practical and important applications. This project report presents the review of question answering from a pedagogical perspective. An overview of the learning goals, description of methods for question answering including current state-of-the-art methods, evaluation metrics and the current research progress in this field are given. In addition, lecture slides and an assignment for the question answering topic are provided. Our aim is to provide a comprehensive overview of the field of question answering theoretically and look for future directions from a research point of view. In conclusion, we provide some challenges we faced during the course of building this project material and suggest options for developing an end-to-end question answering coding project.

1 Introduction

Question Answering (QA) is a discipline within the field of Natural Language Processing (NLP) that deals with building computer systems to answer human questions in a natural language. It is one of the earliest disciplines in the field, reflecting the deep-seated human quest for knowledge, according to Jurafsky [13]. It is also a growing research field with applications in a wide range of contexts, such as in search, dialogue, information extraction, and summarization (reference: Oxford slides). Systems are already in widespread use with key examples such as IBM's DeepQA system Watson in Jeopardy!, voice interfaces like Apple's Siri and Google Search.

QA combines areas in NLP the class has learned throughout the term, yet it also touches on new concepts and application areas. Learning QA would create a more well-rounded understanding of NLP and provide exposure to further research areas. Therefore, we believe it is a worthwhile area to present pedagogically. This document describes how we will be making a small contribution to NLP education by covering the topic of QA. The goal of this pedagogical project is to prepare materials to teach the class a topic that has not been covered. We will first give a broad overview of the different areas in QA to give the class a much needed context of how they are used in research and industry as well as what algorithms are being used. We will touch on Information Retrieval (IR) and Knowledge Based (KB) question answering, including semantic parsing and reading comprehension, with emphasis on newer neural approaches. Then, we will go into the specifics of more niche areas such as Visual Question Answering (VQA) and Conversational Question Answering (CoQA). We will reflect on which sources we used for the project, discuss popular datasets, and demonstrate several methods. Finally, based on these above learning goals for the class, we will provide lecture slides and design an assignment that will test these learning goals.

The remainder of this project report is organized as follows. Section 2 gives some background and applications of QA. Educational materials selected for this project are listed in Section 3. In Section 4 the learning goals are provided. Section 5 elaborates on the lecture plan including methods to be covered, sub areas of QA like VQA and CoQA, research progress and evaluation metrics. In Section 6, the additional teaching materials provided with this report are listed. In Section 7, a list of demos are provided which show the QA systems for a variety of domains. Finally in Section 8, some of the challenges faced are described and we conclude with some suggestions for additional class work.

2 Background and Applications of QA

Question-answering is one of the earliest NLP tasks and is considered a classic problem. Two of the earliest QA systems in the late 1960s and early 1970s answered questions about US baseball statistics and about lunar rocks returned by Apollo missions to the moon [11, 31]. At around the same time, Terry Winograd at the M.I.T. Artificial Intelligence Laboratory developed a system that was able to carry out a dialog with a user about a small world made up of blocks [30]. Since then, QA systems evolved to include many more domains and more modern methods and grew to accommodate internet search engines in the 1990s. Current applications of QA involve tools people use everyday, including Siri, Google Search, and pedagogical tools that answer student questions on reading comprehension tasks [5].

3 Educational Materials Selected for QA

The lecture plan largely follows Chapter 25 on QA from the textbook [13], as it covers the breadth of the topic at a good level of detail. For the more recent research topics like VQA, we used the relevant papers as lecture material. References for any lectures, research papers, webpages, websites, blogs or demos are provided as required in the relevant sections of the report.

4 Learning Goals

As noted above, the lecture plan will follow Chapter 25 of the textbook [13] on QA. The history, methods, applications and research, represent a big part of our teaching plan.

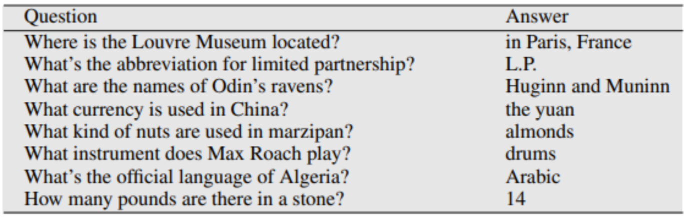
Initially, the emphasis will be on the broad overview and description of algorithms of different information retrieval and knowledge based methods. Many applications of QA in research are based in these two classes of methods so a thorough understanding of them is needed to give students a proper foundation of QA. The methods we will highlight will ideally provide enough information for students to follow along as more state-of-the-art research is enumerated. Demos of QA systems will be provided to show how these methods work. We will operate under the assumption that the students will have already learned neural fundamentals, as a deeper understanding of neural methods will be needed to understand some of the current state-of-the-art works we will present. In addition, other NLP fundamentals including semantic analysis, lambda expressions, and syntactic parsing, are assumed to be already covered as prerequisites. The textbook [13] places QA towards the end as one of the last chapters, indicating that a bulk of the other chapters material is needed to understand this topic. We will take this cue and suggest that a lecture on QA, such as the one planned in this report, should be scheduled towards the end of the term.

To complement the lecture on QA methods, we will next describe the research progress, areas of new development, and future directions in the field of QA. Among them, we will present in more detail two active subfields in QA research. Visual QA (VQA) is a field that combines computer vision, namely image processing and recognition, with NLP methods, both from a machine learning perspective, to create useful applications. This largely involves tasks where a system answers an open-ended, free-form query question on a given image in natural language. Another QA application is CoQA, a dataset crafted specifically for learning conversational question answering. Algorithms using CoQA for learning need to process a complex series of related questions instead of just one so as to hold a natural conversational flow.

To summarize, the lecture will cover the following learning goals:

- Theoretical understanding of established methods as well as advanced methods like VQA used in QA.
- Applications of QA in industry and research.
- Development of research orientation.

This will ensure that the students are well equipped with the necessary tools for QA, to work on future projects in the real world. Also, complementary materials such as more detailed lecture slides and an assignment that reinforces the concepts learned will be provided, as described further in Section 6.



Question	Answer
Where is the Louvre Museum located?	in Paris, France
What's the abbreviation for limited partnership?	L.P.
What are the names of Odin's ravens?	Huginn and Muninn
What currency is used in China?	the yuan
What kind of nuts are used in marzipan?	almonds
What instrument does Max Roach play?	drums
What's the official language of Algeria?	Arabic
How many pounds are there in a stone?	14

Figure 1: Examples of factoid questions and corresponding answers [13].

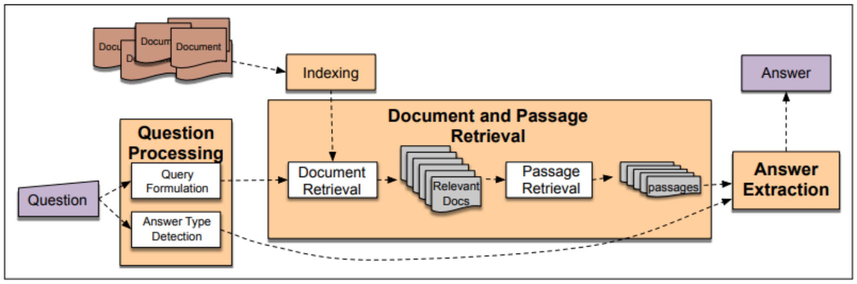


Figure 2: Three phases of IR based question answering [13].

5 Lecture Plan

In this section we provide details of the lecture plan mainly: methods, research progress and evaluation metrics used for QA.

5.1 Methods

5.1.1 Information Retrieval (IR) based Methods

IR based methods are mainly used to answer factoid questions like capitals of countries, birthdays of famous people, etc. An example set of questions is shown in Figure 1. The corpus for extracting the answers for these types of questions is more open-domain, for example the web or a huge set of news articles. There are three main phases in IR-based QA: (1) Question processing, (2) Passage retrieval and ranking, (3) Answer extraction, as shown in Figure 2.

5.1.1.1 Question processing

Question processing is the first step needed to convert the input question into a form which can be used for information retrieval. The most common tasks for this processing are query formulation and answer type detection.

Query formulation can be of different forms. If information is being extracted from the web, the input question can be directly entered into a web search engine. Also, for web search, there might be a need for query reformulation too, where the question is rephrased suitably to get the answer. Other methods like tf-idf cosine matching can be used for smaller documents. In some cases, query expansion is needed, where additional query terms are added, to match particular forms of the answers.

Answer type detection classifies the answer category. For example, whether the answer should be a named entity (person, location or organization), etc. One of the methods is to build hierarchical answer type taxonomies from large lexical databases like WordNet. In this method each question is tagged with a coarse-grained and fine-grained answer type tag. In general, question classifiers are often built from supervised learning, where each answer type is manually labeled for the training database of questions. Both feature-based methods and neural networks can be used for this classification task.

5.1.1.2 Passage Retrieval and Ranking

In the passage retrieval stage, based on the query, a set of documents are selected by the QA system from which the answer can be extracted. These documents are then segmented into smaller passages, for example into sections, paragraphs or sentences. Basic methods pass these passages to the third stage of answer extraction. More sophisticated methods use answer type classification or supervised learning to further select/rank the passages. For supervised learning, measures such as named entities, question keywords, number of n-grams that overlap between the question and the passage, etc. can be used as features. If the corpus is the web, snippets from the web search are used as the passages.

5.1.1.3 Answer Extraction

In this final stage, the answer is extracted from the information sent from the previous two phases, namely question, answer type and passages. It deals with identifying the span of text that constitutes the answer from the passage. This task is commonly called 'span labeling', where the start and end word of the answer are labeled in the passage. In the following sections, previous methods constitute a few of the popular methods used before the advent of neural networks. More modern methods are based on neural networks that provide state-of-the-art results.

Previous Methods:
One of the baseline methods is to use a named entity tagger and find the correct answer from the passage, given the answer type. A limitation of this method is that not all questions have answers that belong to a particular named entity. Hence in general more sophisticated methods are needed to extract answers. The next level of advancement for this task would be to use feature-based supervised learning methods, where the classifier is trained to recognize whether a sentence contains an answer. The answer type as obtained from the question processing stage can be used directly as one of the features. Other features can be constructed for example, by using pattern matching, keyword matching, keyword distance or length of sequence of question that occurs in the answer. Further, methods like n-gram tiling for answer extraction [3, 17] were proposed only for web search QA.

Modern Methods using Neural Networks:
Neural network (NN) models base the QA task assuming that the question and the answer are semantically similar in some way. One of the most common ways to extract answers using neural networks, is in the context of reading comprehension, where the answer is extracted from a given passage. Here the basic underlying approach for most models is as follows. First, an embedding for the question and words of the passage are created. The spans in the passage whose embeddings match closely with that of the question embedding are selected as the answer spans.

One of the most popular datasets used for training NLP QA using NN's is the Stanford Question Answering Dataset (SQuAD) [23] which consists of passages from Wikipedia. The passages have questions associated with them, which were written by humans, and the answers could be found within the passage. Described below are two NN methods for QA. One is based on a bi-LSTM NN model and the other is based on the extremely popular BERT method.

A bi-LSTM-based Reading Comprehension Algorithm:
This method was proposed by Chen et al. in [4]. It is also referred to as the DrQA system. A bi-LSTM based neural network is developed for the question answering task as shown in Figure 3. In this method, a combined (weighted sum) embedding for the question is created by passing the initial GLoVe embeddings [22] of the words in the question through a RNN (such as a bi-LSTM network). This is illustrated in the left side of Figure 3. For the passage each word is represented by an embedding. The input representation for these word embeddings is created by concatenating four components: (1) Embedding of the passage word for example from GLoVe (2) Token features like part-of-speech taggers, (3) exact match features for

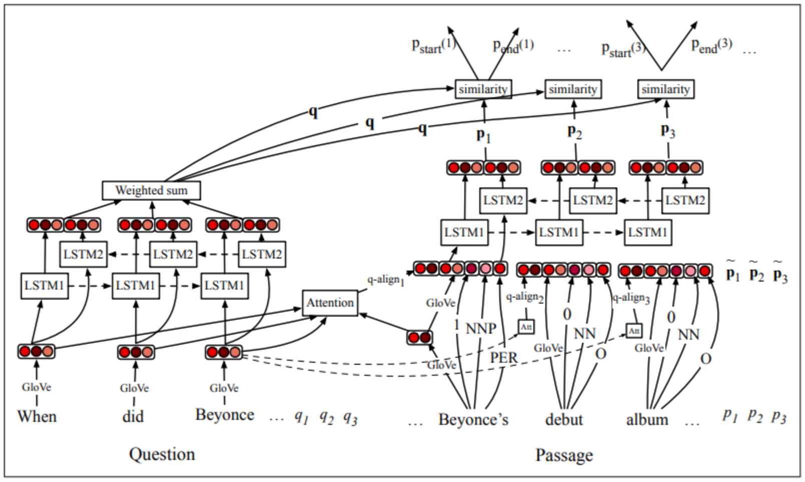


Figure 3: Bi-LSTM based question answering system [13, 4].

example, whether the word is in the question, and (4) adding attention from the question embeddings. This concatenated initial representation is then passed through a bi-LSTM NN to obtain the final embeddings for the passage words as shown in the right side of Figure 3. Given the question and passage word embeddings, the probabilities of the passage embeddings to be the start or end words of the answer span, are obtained by training two separate classifiers (one each for start and end). Here the similarity between the question embedding and the passage word embedding is found usually by using sophisticated similarity functions. The words with the highest probability for the start and end represent the answer span. Also it should be noted that the start word should come before the end word.

BERT-based Question Answering:
BERT stands for Bidirectional Encoder Representations from Transformers and was proposed by [6]. It is a landmark paper in NLP used to pre-train bidirectional representations from text, based on conditioning both left and right context in all layers of the transformer. The BERT architecture for the question answering task is shown in Figure 4. In this model, the two input strings represented as sequences of word tokens, are separated by a SEP token and the BERT contextual embeddings are obtained. In the QA task, the two input strings constitute the question and the passage (or paragraph as shown in Figure 4). From the output tokens for the passage for the BERT model, we can obtain the probabilities for the words to be a start-span word or end-span word for the answer. For this, two new embeddings for S and E are introduced and trained by using some similarity measure to the passage word embeddings, to obtain the start and end probabilities for each passage token. Chapter 25 of the textbook [13] can be referred for further details regarding the downstream fine tuning task.

5.1.2 Knowledge Based Methods

Information can be found in different forms and while it can be found in strings of text, it can also be stored in more structured ways. Knowledge based methods rely on information encoded in these structured formats, usually in a relational database or a Resource Description Framework (RDF) data model. The process by which an answer is given for a question involves mapping the given question to a logical format, such as predicate calculus or a query language like SQL. This process is also known as semantic parsing. There are three main methods in knowledge based QA: rule-based, supervised, and semi-supervised. In this section, we will describe each method and further to illustrate how each is applied, we will give examples found in literature [13].

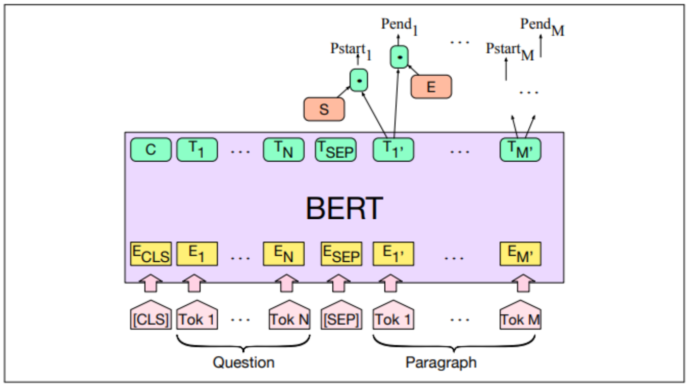


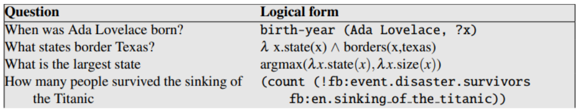
Figure 4: BERT based question answering system [13, 6].

5.1.2.1 Rule-based Methods

Rule based methods use heuristic rules in order to find lexical or semantic information in the question and document. In one of the earliest works of a rule-based method, Quarc [25] uses different rules based on WH question type (who, what, when, where, why) to identify a span of text that holds the answer in a reading comprehension task. Rules can either be lexical (matching words in the question and in the sentence) or semantic (recognizing semantic classes, e.g. HUMAN, NAME). The rules are applied to each sentence of the document and points are awarded on how strongly a rule applies to a sentence. The sentence with the highest score is returned as the answer to the given question. However, the heuristic rules tend to be hand-crafted and may be too expensive to use if the relations are not frequent.

5.1.2.2 Supervised methods

For supervised knowledge based methods, the training data contains questions paired with their correct logical forms [13, 32], as can be seen in Figure 5. A knowledge-based system can then learn from these question-logical form pairs to form generalized rules that can map any new question to its logical form. The learning process involves parsing the questions and aligning the parse trees to the logical form (see Figure 6 top). Then, the mapping of similar but unseen questions to their respective logical forms can be generalized by a rule, like the one in Figure 6 (bottom). This algorithm can be extended in two ways. One, rules can be counted how many times they were used in the parsing of the training data, so probabilities can be used to choose the best parsing for each sentence. Two, multiple relations can be addressed in more complex sentences by breaking down the sentence into smaller more manageable units and then later combining them.



Question	Logical form
When was Ada Lovelace born?	birth-year (Ada Lovelace, ?x)
What states border Texas?	λ x.state(x) $\wedge$ borders(x,texas)
What is the largest state	$\mathrm{argmax}(\lambda x.\mathrm{state}(x), \lambda x.\mathrm{size}(x))$
How many people survived the sinking of the Titanic	(count (!fb:event.disaster.survivors fb:en.sinking_of_the_titanic))

Figure 5: Questions paired with their correct logical forms [13].

5.1.2.3 Semi-supervised and Unsupervised Methods

The problem with supervised methods is that they cannot cover the wide variety of questions that can be asked. Factoid questions tend to be difficult to convert into simple relational forms. Therefore, many methods take advantage of redundancy in the text to structure the logical forms, commonly making use of the vast amount of information found in the web. This can be done in a semi-supervised or unsupervised way. For example, an unsupervised algorithm called REVERB open information extractor [8] creates more com-

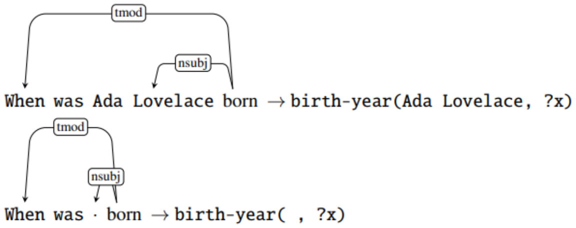
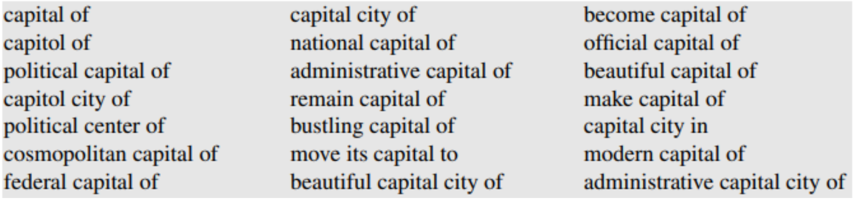

tmod
nsubj
When was Ada Lovelace born → birth-year(Ada Lovelace, ?x)
tmod
nsubj
When was · born → birth-year(, ?x)

Figure 6: Using supervised methods to generate generalized rules (bottom) by learning from the given question (top) [13].



capital of
capitol of
political capital of
capitol city of
political center of
cosmopolitan capital of
federal capital of
capital city of
national capital of
administrative capital of
remain capital of
bustling capital of
move its capital to
beautiful capital city of
become capital of
official capital of
beautiful capital of
make capital of
capital city in
modern capital of
administrative capital city of

Figure 7: Generated list of similar strings that match the Freebase relation of *country.capital* [13].

plicated sets of relations, first creating subject-relation-object triple strings (e.g. "Ada Lovelace", "was born in", "1815") and then combing through a knowledge source like Wikipedia to extract similar strings. Using a predicate that is aligned with the triple string (e.g. a Freebase relation), a set of similar phrases can be found using the cosine distance function comparing the predicate with phrases in Wikipedia. This set of phrases can be consulted to map more varieties of questions to relations. Figure 7 shows the set extracted from the Freebase relation of *country.capital*.

5.1.3 Using multiple information sources: IBM's Watson

Although many methods in QA use either text based or knowledge based methods, as described above, IBM's Watson [9] uses many different resources to answer questions. Watson's DeepQA system that processes the question and answer is split into four main components, as seen in Figure 8. First, question processing handles question parsing, named entity tagging, and relation extraction. Then, it runs focus detection, lexical answer type detection, and question classification. What is interesting to note here is that the set of lexical answer types DeepQA needs for Jeopardy! needs to be far larger than the algorithm described above for factoid questions. This is due to the unique requirements of Jeopardy! as the show asks for a wide variety of question types. In the second stage, the parsed question is combined with knowledge sources, either text documents or knowledge bases, to generate many possible candidate answers. Each candidate answer and lexical answer type are given a vector of scoring features based on various scoring metrics, such as time and space relationships, matching ontologies in WordNet, and retrieving text to see how well an answer can work if substituted with the focus of the text span. Afterwards, any equivalent candidate answers are merged (e.g. JFK-John F. Kennedy or morphological equivalent words), as are their scoring vectors. After a classifier assigns a confidence score based on the vectors to each candidate answer, it then learns the probabilities of the answers being correct and chooses the best answer with the highest probability.

5.1.4 Sub Areas of QA

To give an idea of the variety of directions in QA research, we picked two sub-areas to focus on in our lesson plan. Visual Question Answering (VQA) and Conversational Question Answering (CoQA) are two applications that are more complex than answering simple factoid questions. Delving into these applications can give students an idea of the broad reach of QA in research.

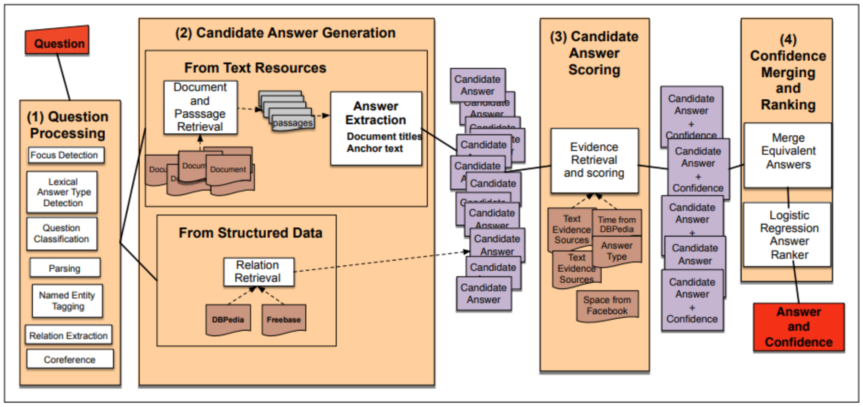


Figure 8: Phases in IBM's DeepQA system Watson [13].

5.1.4.1 Visual Question Answering (VQA)

VQA is a multi-disciplinary area of research, which combines Computer Vision (CV) and NLP. Its main goal is to answer a query question on the image, like how humans see and interpret an image. Figure 9 shows an example of this task. This is an interesting area and has important applications in the real world, where AI systems can extract/provide more information to queries, from images they have access to.

VQA by Agrawal et al. [2] was one of the initial papers proposing free-form and open-ended VQA, where given an image and a question related to that image, the computer system should be able to provide an answer in natural language. This method also takes in diversity knowledge and reasoning unlike the previous methods. In order to achieve this the authors create an abstract scenes dataset which will explore higher level reasoning using VQA. A deep neural network model which combines both image processing and NLP is proposed as shown in Figure 10. The model consists of two channels: (1) A deep Convolutional Neural Network (CNN) adapted from [27] for image processing, which was a breakthrough paper for large-scale image recognition. (2) For NLP for the question, a LSTM based network either with one or two hidden layers is used. The output from the two channels gives a combined image + question embedding which is first passed through a fully connected neural network classifier and then through a Softmax layer to obtain a set of answers. The answer with the highest activation is picked as the final answer. The QA task was performed for both open-ended questions and multiple choice questions. In particular, Figure 10 shows the network for the two hidden layers architecture for the LSTM. This model gave excellent results compared with all the baselines. For interesting further reading in the area of VQA, some of the more recent works that were proposed are [20, 10, 1, 29, 28], which mainly dealt with improvements in the network structure for VQA, for example adding attention mechanisms.

5.1.4.2 Conversational Question Answering (CoQA)

CoQA [24], a Conversational Question Answering dataset developed by the Stanford NLP group, aims to measure how a series of questions can be used for reading comprehension. The motivation behind this was that people tend to not ask just one question to learn about a topic but a series of questions following up on what was asked before. Humans can naturally follow the context of the conversation history even when ambiguity is introduced, but this is a very difficult problem for computers to parse especially because there is a scarcity of datasets containing these ambiguities to train on. Other datasets like SQuAD [23] which has been viewed as a benchmark for reading comprehension has been found to have a narrower set of question types and furthermore, co-references like pronouns and one word questions are virtually non-existent.

CoQA, on the other hand, is filled with these co-references (see Figure 11 where co-references are matched by color). The dataset was developed by having pairs of workers on Amazon Turk asking and answering questions on spans of text from seven different domains. There is now a challenge by Stanford NLP where

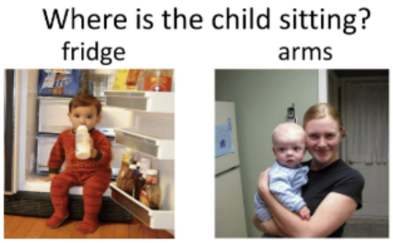


Figure 9: VQA example. (Link to reference webpage.)

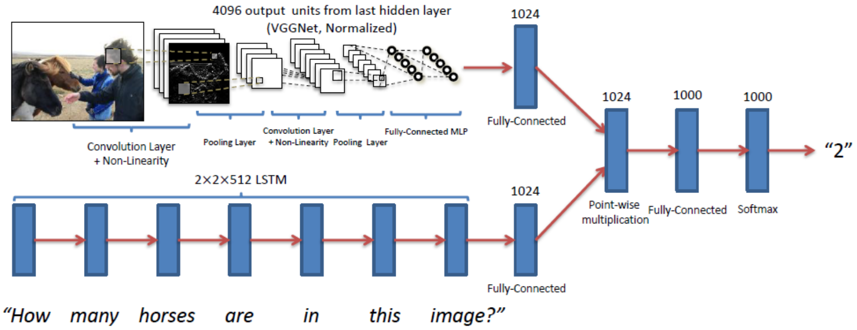


Figure 10: Neural network architecture for VQA [2].

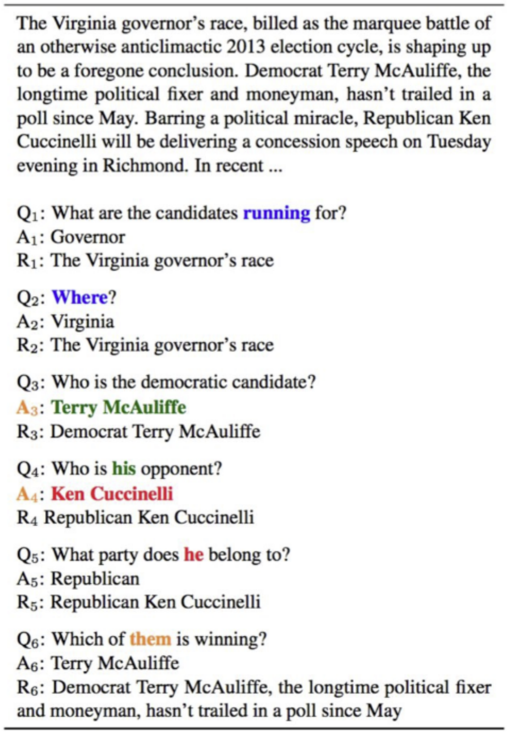
researchers can submit their algorithms that can answer the dataset's conversational questions effectively. Human performance of 88% is the benchmark and currently, there are six that have surpassed it. The top two spots in the leaderboard is a team from ZhuiYi Technology where they used variations of a model they named RoBERTa + AT + KD based on BERT, Adversarial Training, and Knowledge Distillation [12].

5.2 Research Progress (current state-of-the-art)

Overall, progress in QA is similar to many NLP tasks in that it is limited by the datasets available. According to NLP-progress, a website that catalogs NLP research and tracks datasets as well as current state-of-the-art, QA is shown as a robust area of research with at least 25 different datasets across three broad types of applications: reading comprehension, knowledge base, and open domain. For each, we will describe what has been done and what milestones have been reached.

By far the most prevalent in terms of quantity is reading comprehension with about 20 different datasets. There are different types of reading comprehension tasks on various domains, including conversational QA (QUAC, CoQA), Cloze-style reading comprehension (CliCR, CNN/Daily Mail, Story Cloze Test), commonsense inference sentence completion (SWAG, CODAH), and inference from multiple sentences or documents (DuoRC, WikiHop, MedHop). According to their corresponding leaderboards, the state-of-the-art for most datasets involve neural approaches, specifically systems that feature attention mechanisms. One of the more influential systems in QA is the Bi-Directional Attention Flow [26] which in turn inspired the BERT model. Currently, variations of the BERT model have been among the state-of-the-art solutions for many datasets, including CoQA, QUAC, SWAG, and SQuAD 2.0. These include ALBERT (A Lite BERT for Self-supervised Learning of Language Representations [15]) and RoBERTa (A Robustly Optimized BERT Pretraining Approach [19]), both published in 2019.

For knowledge based QA, the development is to tackle semantic parsing with neural and deep learning approaches. The problem with supervised data is that it is very expensive to annotate, especially because human experts are needed to express a question into a query language particular to the schema of a database (reference: Oxford slides). Therefore, more semi or unsupervised methods have been proposed. Of late,



The Virginia governor's race, billed as the marquee battle of an otherwise anticlimactic 2013 election cycle, is shaping up to be a foregone conclusion. Democrat Terry McAuliffe, the longtime political fixer and moneyman, hasn't trailed in a poll since May. Barring a political miracle, Republican Ken Cuccinelli will be delivering a concession speech on Tuesday evening in Richmond. In recent ...

Q_1: What are the candidates **running** for?
A_1: Governor
R_1: The Virginia governor's race

Q_2: **Where**?
A_2: Virginia
R_2: The Virginia governor's race

Q_3: Who is the democratic candidate?
A_3: **Terry McAuliffe**
R_3: Democrat Terry McAuliffe

Q_4: Who is **his** opponent?
A_4: **Ken Cuccinelli**
R_4 Republican Ken Cuccinelli

Q_5: What party does **he** belong to?
A_5: Republican
R_5: Republican Ken Cuccinelli

Q_6: Which of **them** is winning?
A_6: Terry McAuliffe
R_6: Democrat Terry McAuliffe, the longtime political fixer and moneyman, hasn't trailed in a poll since May

Figure 11: Example from CoQA dataset. (Link to reference webpage.)

neural approaches with attention mechanisms have been very popular as well as using machine translation methods. An overview of these methods as shown in the lecture slides will follow closely the Oxford lecture on QA, which includes several works using machine translation attention mechanism [7], semi-supervised training [14], and exploiting the rigid structure of the question to generate the answer [16, 18].

5.3 Evaluation Metrics and Error Analysis for QA

There are several commonly used metrics to evaluate the accuracy of a factoid QA system. One is the mean reciprocal rank (MRR) which takes the mean of the sum of reciprocals of the ranks of the correct answers. This assumes that the system outputs a ranked list of answers for each question. The closer the MRR is to 1, the more accurate the system is for a given dataset. For reading comprehension systems using specialized datasets like SQuAD, the metrics used are either the standard F1 score or an exact match (i.e. accuracy), the percentage of answers that exactly match the gold answer. Other metrics are specific to the peculiarities of the dataset. For example, if the question is presented with a multiple choice list of answers and the task is to choose the correct answer, the evaluation metric could be as simple as the percentage of correct matches. Usually, the "correct" answer that the predicted answer is compared against is human generated. For neural approaches, an evaluation metric more suited for neural methods can be used. For example, using perplexity to assess the quality of the predicted answer by comparing the answer with the test data. For semi-supervised and unsupervised methods, machine translation attention mechanisms are sometimes used. In these cases, the BLEU score (Bilingual Evaluation Understudy) [21] is an option. However, details of all this evaluation metric can be difficult to describe within the time constraints of this lecture.

6 Additional Materials

We will prepare lecture slides for more elaborate, detailed description and easier illustration of the topics with more emphasis on the applications, methods and research work. Much of the material for the lecture slides will be from the text [13] and Oxford lecture (reference: Oxford slides). In addition, to evaluate the effectiveness of the lecture and student's understanding of the material, we will present an assignment. The questions will test not only an understanding of the concepts of QA but also critical thinking skills by asking students to implement small representative QA tasks. In addition, an exercise is given to address basic QA coding tasks using python (a jupyter notebook file is provided for this exercise). Large coding questions will not be asked because we believe that an assignment that asks for large scale implementations will focus more on the implementation details and less on the broader picture of QA. Furthermore, we will assign paper summaries of pivotal works to give students an idea of how research is conducted in this field. The questions with their accompanying solutions and grading scheme are provided as additional materials with this report.

7 Demos

We found and listed several QA systems that can be demonstrated to the class. These offer slightly different interactions for the user. For example, the BIDAF on SQuAD dataset demo has a list of pre-selected context paragraphs to choose from or the user can input a paragraph of their choice. Others feature a fixed context document, such as the Microsoft QA system using a lengthy "Welcome to Canada" booklet of Canadian laws and customs. The questions can be either free-form or fixed.

1. Microsoft QA system
2. BERT QA
3. Google AI
4. VQA
5. Watson visual recognition
6. BIDAF on SQuAD dataset

8 Challenges Faced and Conclusion

In the course of our pedagogical project, we encountered several challenges that kept us from realizing the full potential of our lesson plan. We found that QA systems tend to be very complex with multiple intermediate steps, starting from data extraction to processing and post-processing. But, delving too much into detailed intricacies would take valuable time away from the lecture that could be covered with something more informative. Yet, because QA is conceptually complex especially with the neural approaches (e.g. BERT), a detailed understanding is needed. Therefore, it was difficult to design a lesson plan that would have the appropriate level of detail, and we frequently grappled with the question of what should be left out. Furthermore, building an end to end QA system in Python or other languages is a big undertaking. Many of the code examples found online tend to have no or underdeveloped instructions for building them. Therefore, we had to make do with demonstrating QA systems that were already built and had a public interface. We regret not being able to run some of the newer systems (e.g. BERT models) as they tended to be the most difficult to run.

Though the current project gives a good introduction on the theoretical aspect of QA, if given more time, we would have liked to assign a reasonably sized coding problem to the assignment. To make it more practical, our original intention was to give a coding project to create an end-to-end QA system using one of the methods discussed above. We even consulted project descriptions from other Universities, Carnegie Mellon 11-411 NLP class project and Stanford CS224n class project. However, planning an assignment such as this, requires a lot of time and resources to put together in order for it to be effective pedagogically. However, we did add a basic coding problem in python in the assignment provided, which gives a good introduction to implementing QA.

In conclusion, we have outlined and described a lesson plan on the theoretical fundamentals and applications of QA systems in NLP. Our lesson takes a broad overview style in order to give students sufficient understanding of fundamental methods as well as current progress in QA research. We also delved into specific papers that highlight important methods. We believe that this balanced approach is effective in teaching the introduction to a complex and broad discipline such as QA. We hope that our contributions will guide future instructions on such an important topic.

References

[1] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering, 2017.

[2] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. VQA: Visual Question Answering. In *International Conference on Computer Vision (ICCV)*, 2015.

[3] Eric Brill, Susan Dumais, and Michele Banko. An analysis of the askmsr question-answering system. In *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10*, EMNLP '02, page 257–264, USA, 2002. Association for Computational Linguistics.

[4] Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. Reading wikipedia to answer open-domain questions, 2017.

[5] Chun-Chia Wang, J. C. Hung, Che-Yu Yang, and T. K. Shih. An application of question answering system for collaborative learning. In *26th IEEE International Conference on Distributed Computing Systems Workshops (ICDCSW'06)*, pages 49–49, 2006.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding, 2018.

[7] Li Dong and Mirella Lapata. Language to logical form with neural attention, 2016.

[8] Anthony Fader, Stephen Soderland, and Oren Etzioni. Identifying relations for open information extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, page 1535–1545, USA, 2011. Association for Computational Linguistics.

[9] David Ferrucci, Eric Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya A. Kalyanpur, Adam Lally, J. William Murdock, Eric Nyberg, John Prager, Nico Schlaefer, and Chris Welty. Building Watson: An Overview of the DeepQA Project. *AI Magazine*, 31(3):59–79, Jul. 2010.

[10] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V in VQA Matter: Elevating the role of image understanding in visual question answering, 2016.

[11] Bert F. Green, Alice K. Wolf, Carol Chomsky, and Kenneth Laughery. Baseball: An automatic question-answerer. In *Papers Presented at the May 9-11, 1961, Western Joint IRE-AIEE-ACM Computer Conference*, IRE-AIEE-ACM '61 (Western), page 219–224, New York, NY, USA, 1961. Association for Computing Machinery.

[12] Ying Ju, Fubang Zhao, Shijie Chen, Bowen Zheng, Xuefeng Yang, and Yunfeng Liu. Technical report on conversational question answering, 2019.

[13] Dan Jurafsky and James H. Martin. *Speech and Language Processing, 3rd ed. draft.* 2019.

[14] Tomáš Kočiský, Gábor Melis, Edward Grefenstette, Chris Dyer, Wang Ling, Phil Blunsom, and Karl Moritz Hermann. Semantic parsing with semi-supervised sequential autoencoders, 2016.

[15] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations, 2019.

[16] Chen Liang, Jonathan Berant, Quoc Le, Kenneth D. Forbus, and Ni Lao. Neural symbolic machines: Learning semantic parsers on freebase with weak supervision, 2016.

[17] Jimmy Lin. An exploration of the principles underlying redundancy-based factoid question answering. *ACM Trans. Inf. Syst.*, 25(2):6–es, April 2007.

[18] Wang Ling, Edward Grefenstette, Karl Moritz Hermann, Tomáš Kočiský, Andrew Senior, Fumin Wang, and Phil Blunsom. Latent predictor networks for code generation, 2016.

[19] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019.

[20] Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. Hierarchical question-image co-attention for visual question answering, 2016.

[21] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics.

[22] Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics.

[23] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. *CoRR*, abs/1606.05250, 2016.

[24] Siva Reddy, Danqi Chen, and Christopher D. Manning. CoQA: A conversational question answering challenge. *CoRR*, abs/1808.07042, 2018.

[25] Ellen Riloff and Michael Thelen. A rule-based question answering system for reading comprehension tests. In *Proceedings of the 2000 ANLP/NAACL Workshop on Reading Comprehension Tests as Evaluation for Computer-Based Language Understanding Sytems - Volume 6*, ANLP/NAACL-ReadingComp '00, page 13–19, USA, 2000. Association for Computational Linguistics.

[26] Min Joon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. Bidirectional attention flow for machine comprehension. *CoRR*, abs/1611.01603, 2016.

[27] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2014.

[28] Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers, 2019.

[29] Damien Teney, Peter Anderson, Xiaodong He, and Anton van den Hengel. Tips and tricks for visual question answering: Learnings from the 2017 challenge, 2017.

[30] Terry Winograd. *Procedures as a representation for data in a computer program for understanding natural language*, 1971.

[31] William A. Woods. Lunar rocks in natural English: Explorations in natural language question answering. 1977.

[32] Luke S. Zettlemoyer and Michael Collins. Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. In *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence*, UAI'05, page 658–666, Arlington, Virginia, USA, 2005. AUAI Press.