IIT DELHI

MASTERS THESIS

Question Answering: AI2 Reasoning Challenge

Author:
Harish Chandra Thuwal

Supervisor:
Dr. Maya Ramanath

A thesis submitted in fulfillment of the requirements for the degree of Masters of Technology

in the

Data Analytics and Information Retreival Group
Department of Computer Science and Engineering

June 25, 2019

CERTIFICATE

This is to certify that the thesis titled **Question Answering: AI2 Reasoning Challenge**, being submitted by **Harish Chandra Thuwal**, is a record of bona fide work carried out by him under my guidance and supervision. The work presented in this thesis has not been submitted elsewhere either in part or full, for the award of any other degree or diploma.

Dr. Maya Ramanath
Department of Computer Science and Engineering
IIT Delhi

June 25, 2019

Abstract

Question Answering: AI2 Reasoning Challenge

by Harish Chandra Thuwal

Question Answering is one of the trending and growing research field worldwide. *ARC* is a new and exciting challenge (March 2018) in this field proposed by the Allen Institute of AI. The task of this challenge is to create a system that can answer grade school level multiple choice science questions.

The work done in this thesis can broadly be divided into two parts. We start of by focusing on developing a non-neural model to solve this challenge. We analyze the then SOTA neural model and based on the analysis we propose a non-neural graph based method using stanford's openIE. This algorithm surpasses all other non-neural methods along with the neural model that we started with. Then we move on to propose another graph based algorithm using stuffIE.

The second part would encompass the analysis and annotation of the questions with the type of reasoning that is required to answer those questions. We successfully completed the manual annotation of all the questions. The analysis of the annotations gives us insight into what type of reasoning is needed to solve the questions.

Acknowledgements

First and foremost, I would like to thank my mentor Dr. Maya Ramanath for giving me the opportunity to work under her supervision. Without her guidance and supervision this work would not have been possible. I would also like to thank my friends and family for their constant moral support and motivation.

I would like to express my gratitude towards the helpful folks online (StackOverflow, Reddit etc.) This work would've moved considerably slowly had it not been for them.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Question Answering

In this information age as the users find it difficult to navigate the large amount of wealth of on-line information, the need for automated question answering system becomes more urgent. A system that can allow a user to ask a question in everyday language and receive an answer quickly and succinctly, with sufficient context to validate the answer. Question Answering systems or QASs address this problem.

Question Answering is a computer science discipline, which focuses on building systems that can answer natural language questions posed by humans. A computer understanding of natural language consists of the capability of a program system to translate sentences into an internal representation so that this system generates valid answers to questions asked by an user. Valid answers here mean answers relevant to the questions posed by the user.

It encompasses several other computer science fields such as **Natural Language Processing(NLP)**, **Information Retrieval(IR)** and **Information Extraction(IE)**. Based on what is answered, Question Answering systems have a lot of applications like online examination systems, document management, document classification, human interaction with computers, information extraction from documents and many more.

The upcoming section attempts to give a brief description of different types of Question Answering Systems (QASs).

1.1.1 Types of QA Systems

Following are some of the classification based on different criteria.

- **Application Domains**

  Restricted domain and Open domain QASs are the two categories on the bases of the application domain from which the questions are asked by the users. Repository of questions is limited in restricted domain unlike open domain questions. Different techniques are required to answer restricted domain questions which rely on domain specific ontology and terminology unlike open domain questions which rely on general ontology and world knowledge to get final answer. Google search for example is an open domain system where any user can search for anything. A College Library's search engine on the other hand is a restricted domain system.

- **Types of data source**

  On the basis of the types of data present in the source text. The different categories are:
  (1) structured data source (Database)
  (2) semi-structured data source (XML)
  (3) unstructured. (article, book, reports)

- **Form of answer generated by QAS**

  Answers are presented to the users in various forms that can be extracted as text snippet taken from source documents or generated answers. The form of answers generally depends upon users' question. Generally, the factoid or list questions have answers in the form of sentences. Causal, hypothetical questions have answers in the form of passages. Confirmation questions have generated answers in the form of either yes or No. Some Opinionated questions have answersin the form of ratings. Dialog questions have short dialog answers.

– **Selecting one option from MCQ**

In these cases the answer returned by the QAS is the label of one of the options provided in Multiple choice Questions. ARC [1] and SciQ [2] are examples of this type. The supporting evidence varies from a single huge corpus for each question to a per question paragraph with supporting evidence for the correct answer.

– **Reading Comprehension**

Most current question answering datasets frame the task as reading comprehension where the question is about a paragraph or document and the answer often is a span in the document. RACE [3] and SQuAD [4] are two examples for this type.

– **Conversational Answers**

Conversational Question Answering Systems need to have a dialog or response to the conversation as the answer. For example CoQA is a large-scale dataset for building Conversational Question Answering systems. The goal of the CoQA challenge is to measure the ability of machines to understand a text passage and answer a series of interconnected questions that appear in a conversation.

In this thesis we'll deal with the ARC or AI2 reasoning challenge which is restricted to the domain of Grade school level science and consist of Multiple choice questions. Details of the ARC are explained in the next section.

1.2 ARC

AI2 reasoning challenge (ARC) is a question answering task proposed by Clark et al. [1]. In this challenge the aim is to create a Question Answering system that can answer the questions in the ARC Dataset.

1.2.1 Dataset

Questions

The ARC dataset consists of 7787 4-way multiple choice science questions. The questions are natural, grade-school science questions which were prepared for human tests. The questions are drawn from a variety of sources and are authored by examiners of 3rd to 9th grade.

The questions are categorized in to two categories using two baseline solvers. They are briefly explained ahead.

- **Baseline Solvers**
  - **Information Retrieval (IR) Solver** [5] - The IR solver uses elasticsearch to find the confidence that a question-option pair is present explicitly in the corpus. The answer is the option with the highest confidence.
  - **The Pointwise Mutual Information (PMI) Solver** [5] - It uses PMI to measure the degree to which the parts of a question are associated to the parts of an option. Average PMI is calculated over all pairs n-grams of question and option. The option with highest average PMI is selected.
- **Types of Questions**
  - **Easy Set** - The questions which can be answered either of the two baseline solvers using their default Waterloo corpus are labelled Easy. A total of 5197 questions belong to this category. For example:
    Which property of air does a barometer measure?
    (A) speed (B) pressure [correct] (C) humidity (D) temperature
  - **Challenge Set** - The questions that both the baseline solvers fail to answer correctly are considered "hard" questions and are a part of the Challenge Set. This process basically removes the factoid

questions. This set contains the remaining 2590 questions. For eaxmple:
Which property of a mineral can be determined just by looking at it?
(A) luster [correct] (B) mass (C) weight (D) hardness

Note: In this thesis the focus is restricted to the Challenge Set Questions.

Corpus

The ARC Corpus is a large collection of 14M science related sentences mined form the web. The sources include science-related documents from net, wiktionary, articles from wikipedia that were tagged science etc. The paper [1], based on an informal, sampled analysis concludes that much of the knowledge required to answer 95% of the Challenge questions is present in the corpus. Though this knowledge might not be present explicitly but it is indeed present.

1.2.2 Performance Evaluation Criteria

Both the Easy and Challenge set are already divided into Train, Dev and Test sets.

	Challenge	**Easy**	**Total**
Train	1119	2251	3370
Dev	299	570	869
Test	1172	2376	3548
Total	2590	5197	7787

TABLE 1.1: Division of ARC Questions in each category

The performance of a QA system on ARC Challenge set is measured by the score 1.2.2 it obtains on the 1172 Questions of Challenge Test set.

Scoring Criteria

- If a system chooses the correct answer it receives **1** point.
- If a system reports multiple answers (say k):
  - If the k answers contain the correct answer than the system receives $1/k$ points.
  - If none of the k options is correct the systems get zero points.

- Final score of the system is the average score it obtains on all the Challenge Test set questions (reported as percentage.)

Chapter 2

Literature Survey

This section contains the description and score of the other models. Since the challenge is relatively new (2018) most of these models were submitted recently. I have divided the models in to two categories based on whether they use some form of Deep Neural Network or not.

2.1 Neural Solvers

This section contains brief explanation of the Neural models:

1. **Decomposable Attention (April 2018) 24.34%**

   Given a sentence pair (**a**, **b**) Decomposable Attention Model proposed in 2016 by Parikh et al. [6] predicts whether **b** is entailed by **a**, **b** contradicts **a** or whether they have a neutral relationship. Clark et al. [1] adapted this model in the following manner.

   - For each question q_i and its corresponding options o_{ij} create hypothesis h_{ij}.
   - Use h_{ij} as query to retrieve sentences from corpus using elasticsearch. Let the retrieved sentences be t_{ij}
   - Calculate the entailment score between h_{ij} and t_{ij}.
   - Report o_i with highest entailment score as the answer.

2. **BiDAF (April 2018) 26.54%**

   BiDAF or Bi-Directional Attention Flow for Machine Comprehension is a span prediction model proposed by Seo et al. [7] in 2017. Given a question and a comprehension the model selects a span from the comprehension as the answer. Clark et al. [1] adapted this model by creating a comprehension c_{ij} using the sentences retrieved by elasticsearch for the hypothesis h_{ij}. Then they apply BiDAF to find

a answer span for q_i in c_{ij}. The option o_{ij} that maximally overlaps with its span is reported as the answer.

3. **DGEM (April 2018) 27.11%**

   Similar to Decomposable Attention model, Clark et al. [1] adapted the DGEM or Decomposable Graph based Entailment Model (Khol et al., 2018) [8]. DGEM uses a proprietary parse alongside OpenIE. Without the proprietary parser its score drops to 26.41%.

   Note: DGEM was the SOTA model at the time of the beginning of this work and therefore is used in the analysis.

4. Some other models that came in the later half of 2018 are

   - **KG2: 31.70%** [9]

     Based on Contextual Knowledge Graph Embeddings.
   - **TriAN + f(dir)(cs) + f(ind)(cs) 33.39%** [10]

     Used ConceptNet [11] to improve the performance of TriAN model.
   - **BiLSTM Max Out 33.87%**

     Adapted the BiLSTM Max out model from Conneau, A. et al. (2017) [12].
   - **Reading Strategies 42.32%** [13]

     Fine tuned OFT (Open AI Fine Tuned Transformer) (Radford et al., 2018) [14] using reading strategies.

5. When **BERT** (Devlin et al., 2018) [15] arrived in October 2018 it improved nearly all nlp benchmarks. Hence the top models in ARC Challenge use BERT.

   (a) **BERT MRC Transfer (Dec 2018) 44.62%** [16]

   - Pre-train BERT On RACE dataset.
   - Finetune on ARC dataset.

   (b) **QA Transfer (Dec 2018) 53.84%** [17]

   Reading Strategies over BERT

   (c) **MultiTask BERT (April 2019) 48.29%** [18]

   - Training Procedure same as BERT MRC.
   - Trained on ARC-Challenge, ARC-Easy and OpenbookQA simultaneously.

- Different Hyperparameters.

(d) **BertAristoV7 (June 2019) 57.76%** [19]

- Start with BERT-large-cased using whole word masking.
- Fine-tune on RACE dataset.
- Fine-tune on OtherQA datasets.
- Fine-turne on ARC-Challenge train.

2.2 Non Neural Solvers

1. **Random Baseline 25.02%** - All choices are predicted as equally correct.
2. **Table ILP 26.97%** Clark et al. [1] adapted the TableILP [20] approach proposed in 2016.

It's evident that the current SOTA models are all building upon the power of BERT/GPT, which are deep neural networks. On the other hand the highest score achieved by a non-neural method (No embeddings, No Neural Network) is a pretty low 26.97%.

Though DNN based models achieve significantly higher score than non neural approaches, their true gain is however inflated with respect to the computational cost and long training time and often are used as a black box. **So, In this thesis the focus is on creating non neural Graph based approaches and analyzing the questions and the knowledge required to answer those questions.**

Chapter 3

Analysis of DGEM and Corpus

Since DGEM Model was the SOTA model at the beginning of this project it was analyzed to look for bottlenecks that might be leading to low scores.

3.1 Is the ARC Corpus Sufficient?

Although the paper (Clark et al., 2018) [1] based on an informal, sampled analysis concludes that much of the knowledge required to answer 95% of the Challenge questions is present in the corpus. We look at the affect of adding additional resources alongside ARC corpus and see if the model performance improves.

3.1.1 Adding Additional Corpora

Since the ARC Corpus already contains data scraped from wikipedia pages we add the following resources alongside the existing corpus:

1. **WebChild Database** [21]

   It is a large collection of commonsense knowledge, automatically extracted and disambiguated from Web contents. It contains triples that provide various relations between words and their meanings. These triples were converted into sentences and consolidated into a single corpus file. A few examples of the conversion of different types of relations are

   - **meaning** - *word is meaning*
   - **part of** - *word-a is part of word-b*
   - **has substance** - *word-a has substance word-b*
   - **is member** - *word-a is member of word-b*
   - **comparative** - *car is faster than bike*

- **property** - *plant has green color*

2. **NCERT Science Text Books** [22]

   Downloaded the Science textbooks of class 6th till 10th. The downloaded pdfs were then converted to text, clean and consolidated into single file.

3.1.2 Effect of the Additional Corpora

Since the DGEM uses a proprietary parser we use publicly accessible openIE based implementation of DGEM (which gives a slightly less score) which should server the purpose of analyzing the affect of corpus on the DGEM model pipeline.

Corpus Used	DGEM OpenIE score
ARC	26.41
ARC + NCERT	26.20
ARC + NCERT + WebChild	26.41

TABLE 3.1: Performance of DGEM (OpenIE) on Challenge Test set using various corpuses

As its clear from the table 3.1 that adding the new corpora had no positive effect on the performance of the DGEM model. To further analyze the problem we manually looked at the questions that were answered incorrectly by DGEM using any corpus combinations. One of those is as follows:

Question: *An astronomer observes that a planet rotates faster after a meteorite impact. Which is the most likely effect of this increase in rotation?*
Correct Option: *Planetary days will become shorter.*
Some sentences from corpus:

- *...**Earth's** rotation was changed by the earthquake to the point where the days are now 1.8 microseconds shorter...*
- *... changed the **earth's** rotation, making the **day** a bit **shorter** than usual...*
- *... **earth** is the 3rd **planet** from the sun...*

Its clear from the above example that even though the corpus does contain the information necessary to answer the question, the DGEM model was unable to answer the questions correctly. Therefore we move on to analyzing the DGEM pipeline itself.

3.2 Bottleneck in DGEM Pipeline

Before we go forward, here is a brief overview of the steps in the DGEM pipeline.

1. Create Elasticsearch index for each corpus.
    - Split text on anything that is non alphanumeric.
    - Perform conversion to lowercase, stopword removal, stemming.
2. For each question q_i and option o_{ij}.
    - Use q_i and o_{ij} as query to the elasticsearch index.
    - Top 8 hits are called support sentences. Say S_{ij}.
3. Combine q_i and o_{ij} to create h_{ij} (Hypothesis) by replacing the "wh" words in the question with the option.
4. Ask DGEM model (trained on manually curated entailment dataset) How much each of the support sentence (S_{ij}) entails the hypothesis (h_{ij}).
5. Return the maximum entailment score for each option.
6. Predict the option with maximum entailment score.

We manually analyzed each step of the DGEM pipeline in the cases where the model predicted wrong answers. We found out that the support sentences being extracted from the corpus by the elasticsearch were of poor quality. In some cases the top scoring support sentences for each option were the same while in other cases the support sentences were not even sentences "just irrelevant words". For e.g. for the question *22 Sandy is conducting an investigation to find out which food his dog likes best. Which is the manipulated variable in his investigation?*, the top scoring support support sentence for each option was "his". As the extraction of the support sentence is the very frst step in the DGEM pipeline, failure in retrieving good support sentences renders the model useless.

Based on the above observations and the fact that we are focusing on a non-neural approach (mentioned at the end of literature survey), we purpose a simple graph based algorithm in the next chapter.

Chapter 4

Graph based scoring algorithm

As concluded from the analysis on the previous subsection elasticsearch was unable to extract good quality support sentences resulting in lower score. We therefore instead of extracting support sentences and then working on them attempt to use the entire corpus.

4.1 Approach I: Using Stanford OpenIE

The Algorithm can be divided into four major components as follows:

1. Hypothesis Generation.
2. Running Stanford openIE.
3. Create Knowledge Graph.
4. A scoring Function.

The following four sections will delve into each components one by one.

4.1.1 Hypothesis Generation

For each question (q_i) option (o_{ij}) pair, a hypothesis (h_{ij}) is generated using the following method.

Note: A *wh-word* refers to English words used to introduce questions and relative clauses. The main *wh-words* are why, who, which, what, where, when, and how.

- If a *wh-word* is found:
    1. Replace it with o_{ij}.
    2. Replace the last question-mark(?) with a full-stop(.).

3. Remove the keyword "of the following" if present immediately after the *wh-word*.
4. For example:
   $\mathbf{q_i}$: *... which of the following force is responsible for...*
   $\mathbf{o_{ij}}$: *Electromagnetic*
   $\mathbf{h_{ij}}$: *...Electromagnetic force is responsible for....*

- If no *wh-word* is found and the question ends without a period or question-mark then introduce the option at the end. For example:
  $\mathbf{q_i}$: *The gravitational force exerted by an object depends on its*
  $\mathbf{o_{ij}}$: *Mass*
  $\mathbf{h_{ij}}$: *The gravitational force exerted by an object depends on its Mass*.

- If all else fails, assume "this?" indicates the place for option. For example:
  $\mathbf{q_i}$: *Virtually every task performed by living organisms requires this?*
  $\mathbf{o_{ij}}$: *Energy*
  $\mathbf{h_{ij}}$: *Virtually every task performed by living organisms requires Energy*.

Using the above method for every question option pair a hypothesis is created. These hypothesis will be used in the next step. Assuming we have 4 options per question, number of hypothesis is equal to 4 times the number of questions.

4.1.2 Stanford OpenIE

Open information extraction (open IE) refers to the extraction of relation triples, typically binary relations, from plain text, such as (Mark Zuckerberg; founded; Facebook). The relations extracted are open domain relation triples, denoting a subject, a relation, and the object of the relation. Stanford OpenIE is a part of Stanford CoreNLP [23] package which performs openIE.

Consider the following examples:

- **Sentence1**: *The U.S. president Barack Obama gave his speech on Tuesday and Wednesday to thousands of people.*

  Triplets1:

  – Stanford OpenIE

    1. *(U.S. president Barack Obama; gave; his speech)*

  - Stanford OpenIE with coreference resolution
    1. (Barack Obama; is; president)
    2. (Barack Obama; is; U.S.)
    3. (U.S. president Barack Obama; gave; U.S. president Barack Obama speech)

- **Sentence2**: *Jack and Jill visited India, Japan and South Korea.*

  Triplets2:

  1. (Jill; visited; South Korea)
  2. (Jack; visited; India)
  3. (Jill; visited; India)
  4. (Jill; visited; Japan)
  5. (Jack; visited; Japan)
  6. (Jack; visited; South Korea)

We run the stanford openIE on the entire corpus and each hypothesis to obtain corresponding set of triplets. These triplets will be used in the next step to create graphs.

4.1.3 Creating Graph

Each triplet (subj; relation; object) is considered as a directed edge from "subj" node to "object" node with "relation" as the edge label.

Examples:

- If the triplet is of the form *(Barack Obama; gave; speech)*. This will result in the following edge

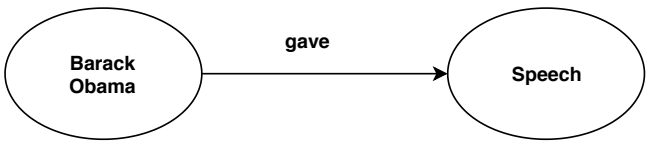

FIGURE 4.1: Example of an openIE graph edge

- For the triplets of the second example in the previous subsection (Triplets2) the graph would look as shown in Fig 4.2

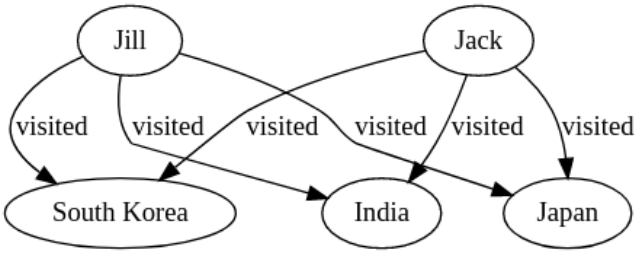

FIGURE 4.2: Example of a connected openIE graph

- The resultant graph might not always be connected. The above example results in a connected graph because the graph corresponds to a single sentence. In case of multiple sentences the resultant graph will most likely be disconnected. For example the graph corresponding to the first 100 triples that were extracted from the downloaded corpus is shown in Fig 4.3

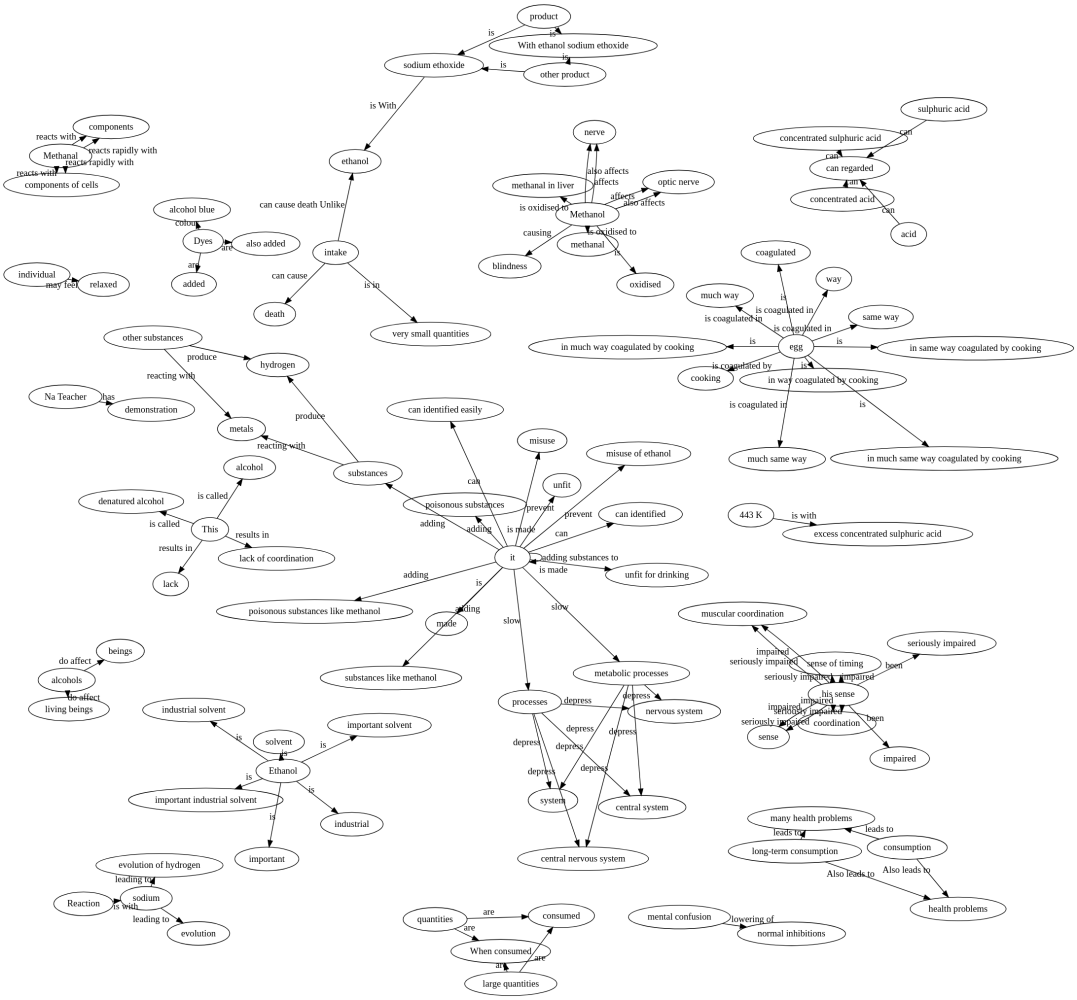

FIGURE 4.3: Example of a disconnected openIE graph

In this way a single big corpus graph C and corresponding to each hypothesis h_{ij} a hypothesis graph H_{ij} are created. These graphs are used in the next step to calculate a score for each option of a question.

4.1.4 Simple Scoring Algorithm

Given a Corpus Graph C and a Hypothesis Graph H_{ij}. The score(C, H_{ij}) is the sum of the following:

1. Fraction of nodes of H_{ij} that are also present in C (Exact Match).
2. For each edge *a—p–>b* in H_{ij} (where *a* and *b* are nodes and *p* is the edge label)
    - If an edge *a—q–>b* is present in C, score for the edge *a—p–>b* is defined as average of *score_edge_labels(p, q)* for all such q.
    - *score_edge_labels(p, q)* can be any of the following:
        - (F) Fraction of words of p that are in q.
        - (E) 1 - editdistance(p, q)
        - (JC) 1 - Jaccard Similarity (p and q are sequence of characters).
        - (JW) 1 - Jaccard Similarity (p and q are sequence of words).

4.2 Approach II: Using StuffIE

Similar to Approach I (Section 4.1) this can also be divided into four major components:

1. Hypothesis Generation.
2. Running StuffIE.
3. Create Knowledge Graph.
4. A scoring Function.

The process of Hypothesis Generation is exactly same as the Approach I. So, in the following sections we'll discuss the remaining three steps.

4.2.1 StuffIE

StuffIE (Prasojo et al., 2018) [24] is a more-informative OIE (Open Information Extraction). Its fine grained and facet-centric.
The main advantages of stuffIE over Stanford openIE are:

1. It extracts nested relations by capturing the various links between facts and their facets in a fine-grained way.
2. It avoids having relations with complex arguments not to loose information about facets.
3. It captures the semantic role of facets.

StuffiE relation format

In ths section we briefly explain the StuffIE relation format. The relation given by StuffIE as output is of the format **< *i, mr, f* >**

- ***i***: Unique identifier of the relation. For e.g "#1.4".
- ***mr***: Main relation is a triplet describing the main fact. Its of the form **< *s, p, o* >** where each element of the triplet denotes **s**ubject, **p**redicate and **o**bject respectively.
- ***f***: A set of extra details about the main fact. Its of the form **< c_j, f_j >**. Here f_j is one of the extra details about the main fact and c_j is the facet connector connecting the main fact to the extra detail. For example c_j can be "of", "at", "on", "in case of" etc.

- ***s***, ***o*** and f_j are either noun phrases or reference ot other relations.
- ***p*** is either a verb phrase or clause connector if ***s*** or ***o*** is a reference to other relations.
- A verb/connector can be synthesized if the explicit one cannot be found.
- Whenever a verb or a connector is expected to be there but can neither be found or synthesized, <_> is used to represent an empty placeholder for them.

For example Consider the following sentence: *"Of course, for many in the media, hydrogen sulphide delivery helps prevent disease damage in cells in certain disease models will always be trumped by farts cure cancer when it comes to headlines."*

On running stuffIE over this sentence we get the following relations.

1.13: hydrogen sulphide delivery; helps; #1.29;
<_1>; for many in the media;
<_2>; Of course;
1.14: <_3>; prevent; #1.26;
1.26: disease damage in cells in certain disease models; will be trumped by; farts;
<_4>; always;
1.29: #1.14; cure; cancer;
when; #1.33;
1.33: it; comes to; headlines;
1.21: models; <be>; disease damage in cells in certain disease;

We run StuffIE on the entire corpus and each hypothesis to obtain corresponding relations. These relations are used in the next step to create graphs.

Creating Graph

Based on the format of the stuffIE relations we decided that in the graph:

1. subject, predicate, object, facet_connector, and facet are all represented as a **node**.
2. Every main relation of the form **< *s*, *p*, *o* >** indicates two edges: one from *node(s) –> node(p)* and another from *node(p) –> node(o)*.

3. We'll associate the facet to the predicate **p** of the main relation. So now the Facet **< c_j, f_j >** can be written as a triple: **< p, c_j, f_j >**. This now can be represented in the same way any main relation is represented (Point 2).

4. *node("node_phrase)* contains the "node_phrase" and a list of phrases of other nodes to which it is connected. "node_phrase" can be a noun_phrase or a verb_phrase or a clause_connector.

5. Graph stores a dictionary (key-value pairs) where the key is the "node_phrase" and the value is the corresponding node("node_phrase") object.

An Issue that arises is that many verb_phrases or facet_connectors that cannot be inferred are replaced with <_> and this leads to many unnecessary edges as "<_>" generated in different relations are considered as the same node by the graph. To deal with this we decided to collapse all the <_> nodes and connect their children to their parents. This results in a much cleaner graph.

For the relations obtained in the example in the previous section the generated graph is shown in Fig 4.4

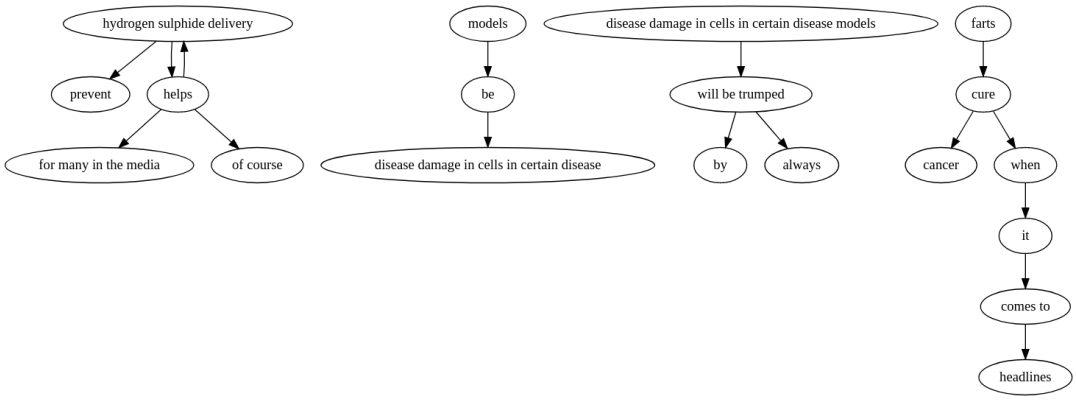

FIGURE 4.4: Example of a StuffIE graph with "_" collapsed

Using the aforementioned representation we create graphs for the corpus and for each hypothesis. These graphs will next be used in the scoring algorithm.

4.2.2 Scoring Algorithm

Given a Corpus Graph (C) and a Hypothesis Graph (H_{ij}). The score(C, H_{ij}) is given as follows:

1. Sort the nodes of the Hypothesis Graph (H_{ij}) in the descending order of depth of the respective subgraph. Let call this

2. For each unvisited node **a** in the sorted nodes of H_{ij} if the node is present in the corpus graph (*C*) then:

   (a) Let the *nbrH(a)* and *nbrC(a)* be the unvisited neighbors of **a** in the H_{ij} and *C* respectively.

   (b) Each node in *nbrC(a)* is a potential match candidate for every node in *nbrH(a)*.

   (c) From the candidates choose the match for each node in *nbrH(a)*.

      i. For each node in *nbrH(a)* say **x** calculate match scores with its candidates. Similar to scoring algorithm of Approach I match scores can be fraction of words, Jaccard Similarity, Edit distance etc..

      ii. For **x** the candidate whose match score is maximum and above the **"match_threshold"** is chosen as a the match say **y**.

      Let

      $depthH_{ij}^{x} =$ depth of Node x in H_{ij}

      $depthC^{y} =$ depth of Node y in C from the last match

      $mscore^{x} =$ match score of x with y

      $tscore =$ total score between C and H_{ij}

      Then

      $$tscore+ = mscore^{x} \times \frac{depthH_{ij}^{x}}{depthC^{y}}$$

      Repeat the step (c) for all the unvisited children of **x** with their candidates set to *nbrC(match)*

      iii. If none of the candidates surpass the match_threshold then update the candidates of x as follows:

```
new_candidates = []
for each in candidate_of_x:
    doe = depth_of_each_from_last_match
    if doe <= depth_threshold:
```

```
                new_candidates.append(children_of_each)
        if new_candidates is not empty:
            Repeat the step (c) for x with
            new_candidates
```

The above algorithm is controlled by three hyperparameters which are as follows:

- **depth_threshold**: Depth from the last match upto which the subgraph in corpus will be explored to find a match for the candidate nodes.
- **match_threshold**: Match score above which a node match is considered potential.
- **beam_threshold**: Since we are searching all the children in the corpus_graph the number of potential candidates will grow exponentially. We need to control this by specify the maximum_number of candidates to search at a time. This is dictated by the beam_threshold.

4.3 Results

4.3.1 Approach 1

Details of the generated corpus graph are shown in the Table 4.1

Method Used	Time Taken	Number of Nodes	Number of Edges	Number of Nodes in the largest Component
Stanford OpenIE	11.39 Hours	21317833	40606342	20605758
Stanford OpenIE + Coreference Resolution	62.8 Hours	22123223	41976390	21236195

TABLE 4.1: Details of corpus graphs generated using approach I

Coreference resolution is the task of finding all expressions that refer to the same entity in a text. For example consider the sentence *"I voted for Nader because he was most aligned with my values", she said*. In this sentence my refers to she and he refers to Nader.

Therefore when we apply coreference resoluton to the corpus, the pronouns would be replaced by the corresponding entity. This should affect the size of the corpus graph but as we can see this is not the case and the differences are not significant. This might be because of the fact there is no structure in the ARC Corpus. The corpus is just a huge collection of random sentences from academic resources and no context exists around sentences.

Method	Score
Approach I (F) + Coref Graph	27.82
Approach I (F)	27.41
DGEM + Proprietary Parser	27.11
Table ILP	26.97
DGEM + OpenIE Parser	26.41

TABLE 4.2: Approach I (F) vs DGEM

It can be seen from the Table 4.2 that the Approach 1 not only scores higher than the best non-neural model of Table ILP [20], it also surpasses the DGEM Model. Approach I (F) stands for using Fraction of common words as the edge scoring criteria mentioned in Section 4.1.4.

Though the overall result seemed to have improved it can be seen from the Figure 4.5 that DGEM is getting a score of 1 more percentage of questions as compared to Approach I (F). Approach I scores a overall higher score because of decrease in incorrect answers. It avoids getting a zero score by predicting a more number of k-way ties.

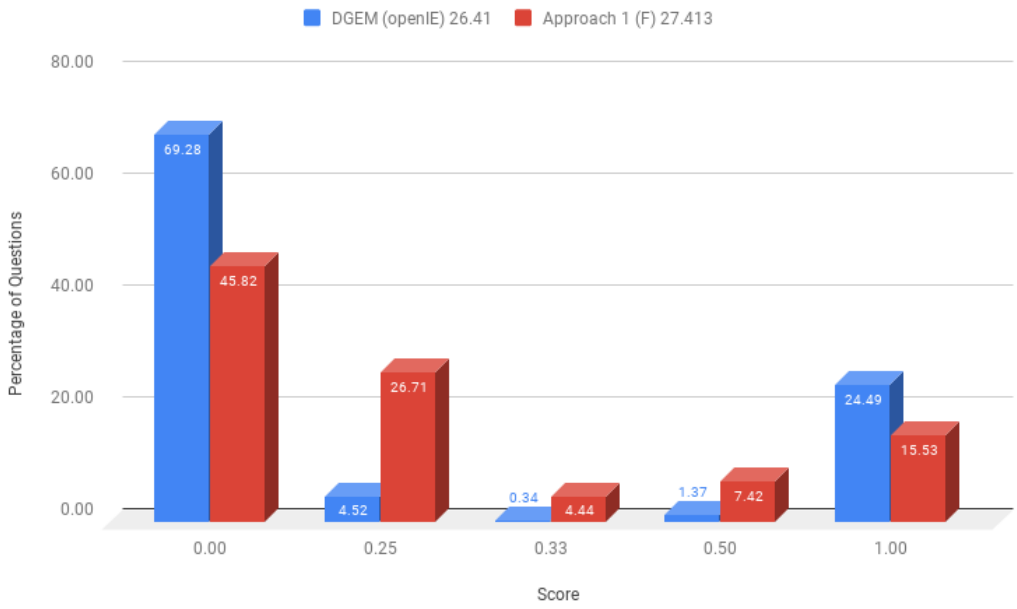

FIGURE 4.5: Percentage of Questions vs Possible Scores
DGEM vs Approach 1

Table 4.3 shows the performance on using different edge scoring methods in Approach 1. All the variations perform above the DGEM baseline and the best performing configuration is almost 2% better than thd DGEM model. **Note** that the DGEM is Neural Graph Entailment based model while our Approach is non-neural and is completely deterministic.

Edge Scoring Method	**Score**
Approach I (JC)	29.0031
Approach I (E)	28.124
Approach I (JW)	27.68
Approach I (F)	27.41

TABLE 4.3: Approach I - Performance of various edge scoring methods.
Refer Section 4.1.4

4.3.2 Approach 2

Number of Nodes	Number of Edges	Size of Largest Strongly Connected Components	Size of Largest Weakly Connected Components
21310962	45913182	3787116	21216503

TABLE 4.4: Details of the corpus graph generated using approach II

Although it appears from Table 4.1 and Table 4.4 that both apporaches created graphs of similar sizes but thats not the case. We need to consider the fact that in the graphical representation of Approach 2 there are no edge labels. That is everything is considered a node. On the other hand in the representation of approach 1 edge labels are not considered nodes. Keeping this in mind we can conclude that Approach 2 (StuffIE) resulted in much more compact representation of the corpus.

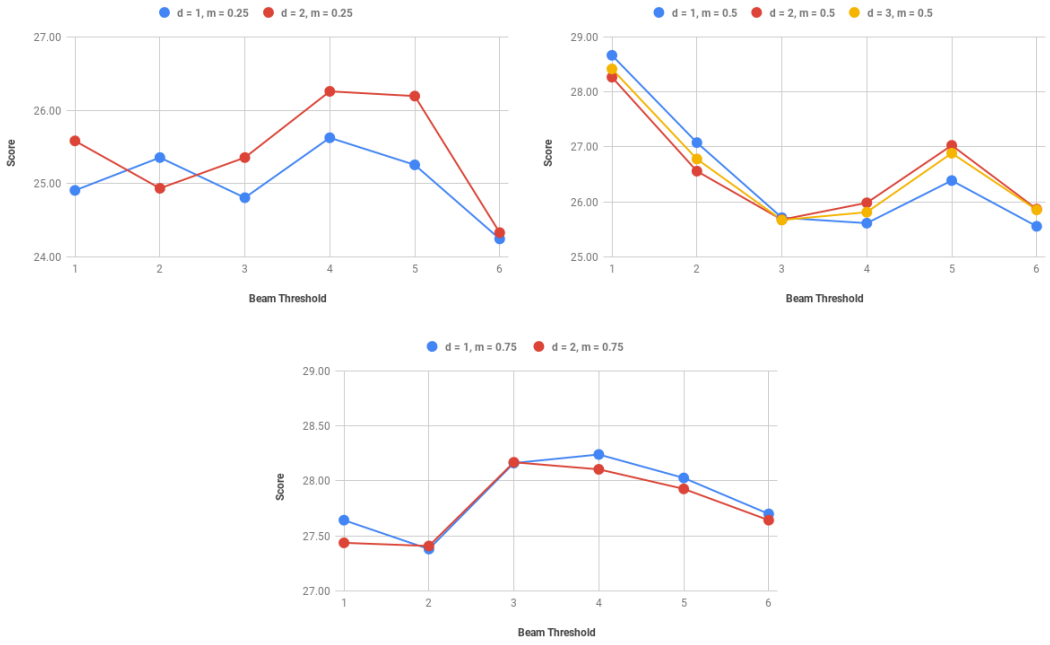

FIGURE 4.6: Variation of score with value of *beam threshold* keeping other hyperparameters constant.

We are using (JC) metric for matching strings as it gave the best performance in the Approach 1. We ran hyperparameter search over the three hyperparameters: *Depth Threshold (d)*, *Match Threshold (m)* and *Beam*

Threshold (b). Refer to the end of the section 4.2.2 for more details about these hyperparameters.

We were very limited in the hyperparameter search space as increasing both b (measure of breadth) and d (measure of depth) blows up the complexity of the algorithm. The maximum score that we were able to obtain was 28.67 which although better than the DGEM and other non-neural models is less than *Approach I* by 0.33.

Figure 4.6 shows the variation of the score with change in beam threshold keeping others constant. When the match threshold is set to low the score also remains low as compared to when the match threshold is high because with lower match threshold incorrect nodes will enter the list of match candidates. Also, at higher beam threshold higher match threshold is required as increasing the beam threshold will bring in more nodes as candidates. A higher match threshold is needed to filter out poor candidates.

Chapter 5

Reasoning Type Analysis of Questions

The paper(Clark et al. (2018)) [1] shows an analysis of 100 Questions with respect to the types of Reasoning required to answer these questions. Boratko et al., 2019 [25] attempts to improve upon this categorization by providing a comprehensive set of categories and creating a separate annotation tool. But even after using 10 different annotators and creating an independent annotation software they only provide annotations for only 196 questions(out of 1119) and that too from the "ARC Challenge Train Set".

So, we use the types of Reasoning mentioned in [25] to manually go through and annotate the 1172 questions of the "ARC Challenge Test set". Using this labelling we can then have a better understanding of the questions.

- what type of reasoning would a student and therefore a QA system require to answer each question in the set correctly?
- Distribution of annotations across the questions. what are the most prominent types of reasoning required to answer the questions?
- Are there any questions that a IE based QA system should not be able to answer?

5.1 Type of Reasoning

There are 9 different Reasoning types that are used to annotate the "ARC Challenge Set" questions. They are:

1. **Question Logic**
    - Questions which only make sense in the context of a multiple choice question.

- That is, if the choices are removed, the question makes no sense. Once the choices are available the question along with the options should be sufficient to answer the question.
- **"Quick Identification"**: If on removing the answer options it becomes nearly impossible to answer the question, then it requires a reasoning of "Question Logic" type.
- **Example**: Which item below is not made from a material grown in nature?
  (A) a cotton shirt
  (B) a wooden chair
  (C) a plastic spoon
  (D) a grass basket

  Here if we remove the options there is no way to be absolutely sure about the answer of the question as it asks us to choose from the options. Even if we try to answer it without the option there is no unique answer because there are many materials that are grown in nature and a large number of items can be made from each of them.

2. **Linguistic Matching**

- Questions that requires itself to be aligned with a sentence or a fact to identify the answer.
- This label often goes with labels like multihop reasoning(4) and causal(3) where the facts are useful only when aligned with the question.
- **Example**: Which of the following best describes a mineral?
  (A) the main nutrient in all foods
  (B) a type of grain found in cereals
  (C) a natural substance that makes up rocks
  (D) the decomposed plant matter found in soil

  Assuming the corpus contains a statement of the form "minerals are found inside rocks" even then we'll need to align the option(C) with it then only we'll be able to say that option(C) is correct.

3. **Causal/Explanation**

   - Assuming that an ideal corpus is available from which facts can be retrieved, the answer can be deduced from a single fact or element and often requires Linguistic matching(2) to align with that fact.
   - Includes single hop causal processes.
   - Includes questions of the form *"What is the most likely result of X happening?"*
   - **Quick Identification**: If only one statement of fact would be required to explain the answer to a 10 year old, then the question can be labelled Causal/Explanation.
   - **Example**: Why can steam be used to cook food?
     (A) Steam does work on objects.
     (B) Steam is a form of water.
     (C) Steam can transfer heat to cooler objects.
     (D) Steam is able to move through small spaces.

     To answer this question the only statement we need is the one that contains the fact that **Steam is used to heat food**.

     Note that one can also see this as a multihop reasoning of the form "steam is used to heat objects", "food is a type of object" and "cooking involves heating". This ambiguity in deciding the label is one of the issues that arise and is discussed later in the section 5.2.

4. **Multihop Reasoning**

   - Assuming that an ideal corpus is available from which facts can be retrieved, the answer can only be extrapolated if at least two or more unique pieces of fact/evidence are available. These facts will often need to be matched linguistically(2) with the questions.
   - Includes multihop causal processes.
   - Includes questions of the form "If X happened what would happen to Y?"
   - **Quick Identification**: You would need at least two distinct statements of fact in order to explain this type of question to a 10 year old.

- **Example**: Which property of a mineral can be determined just by looking at it?
  (A) luster
  (B) mass
  (C) weight
  (D) hardness

  This is clearly a case of multihop reasoning as we need to use multiple facts like "luster involves shine" and "shine can be seen/looked at" to decide that the answer is luster.

5. **Hypothetical**

- Questions that describe an imaginary situation which may not be present in the corpus in any form.
- Questions where you need to apply general abstract facts to a hypothetical scenario mentioned in the question in order to find what would happen in that hypothetical situation. Hence they often require you to do Linguistic Matching(2) and Causal(3) or Multihop(4) reasoning.
- This include the cases where instead of the question each option is a hypothetical situation and you need to choose the correct one.
- A situation/entity that may or not be present as a statement of fact in the corpus is considered hypothetical. For example *"..a gray squirrel gave birth to a ...", "When lemon juice is added to water.."* etc.
- **Example**: If the Sun were larger, what would most likely also have to be true for Earth to sustain life?
  (A) Earth would have to be further from the Sun.
  (B) Earth would have to be closer to the Sun.
  (C) Earth would have to be smaller.
  (D) Earth would have to be larger.

  Here we need to imagine a hypothetical situation where the size of sun had increased and apply our knowledge of affect of size on force of gravitation and radius of the orbit to the situation in order to find the right answer.

6. **Comparison**

   - Questions that directly ask or require you to compare one or more entities.
   - The entities that are needed to be compared can be present in the question or the options or both of them.
   - The comparison need not be along one dimension. You may need to compare multiple attributes of the same set of elements.
   - Includes question of the form *"What is the most likely place to find water?"*. This is because although this question does not contain a word indicating "comparison", we'll still need to compare the options on whether or not they have water.
   - **Example**: Compared to the Sun, a red star most likely has a greater
     (A) volume.
     (B) rate of rotation.
     (C) surface temperature.
     (D) number of orbiting planets

     In this example we need to compare multiple attributes(the options) of the sun and a red star and choose the attribute which is greater for the sun.

7. **Algebraic**

   - If answering it requires you to perform some sort of arithmetic/numerical calculation.
   - The calculation can be anything ranging from primitive arithmetic to solving mathematical equations(For e.g. kinematics equation, Newton's laws).
   - These equations would be retrieved from the corpus. We don't expect the QA system to know all possible equations on its own.
   - This is a particularly interesting category as solving questions of this category would not only require the QA model to extract the appropriate equations but the model would also need to solve those equations or perform arithmetic operations. So a IE based QA system that extracts information from a corpus should not be able to answer these questions.

- **Example**: A 72 W navigation unit on a commercial aircraft has a 24 V power supply and uses 3 A of electric current. What is the electrical resistance of the navigation unit?
  (A) 4 ohms
  (B) 8 ohms
  (C) 13 ohms
  (D) 22 ohms

  To be able to solve this question we first would need to extract the fact that given voltage and current the resistance can be calculated using the formula:
  $$R = \frac{V}{I}$$
  and then actually use this equation and divide voltage by current to get the resistance. Its highly unlikely that a statement with exact set of values of voltage, current and resistance is present in the corpus.

8. **Physical Model**

- Questions that require you to know/understand a physical relationship between the entities mentioned in the question or the options or both.
- Physical relationship include relationship involving position or motion of the elements involved.
- Questions that requires a model of the physical world in order to be answered.
- **Example**: Where will a sidewalk feel hottest on a warm, clear day?
  (A) Under a picnic table
  (B) In direct sunlight
  (C) Under a puddle
  (D) In the shade

  In order to answer this question alongside the fact that sunlight can make stuff hot we also need to have a physical model of the world that sunlight won't be able to reach under a picnic table, under a puddle and inside the shade.

9. **Analogy**

   - Similar to Comparison but here we don't directly compare the entities.
   - Questions in this category usually require you to find an option that follows some scenario in the same way other elements follow another scenario.
   - **Example**: Inside cells, special molecules carry messages from the membrane to the nucleus. Which body system uses a similar process?
     (A) endocrine system
     (B) lymphatic system
     (C) excretory system
     (D) integumentary system

5.2 Results

Before moving on to the results of the annotations I would like to point out some points which are as follows:

- It's difficult to come up with a unique categorization for a question. This is because the labelling depends on the following factors:

  - How is the annotator interpreting the questions and
  - the degree of the subject knowledge or lack thereof. For example I have a better understanding of physics as compared to biology/geography so I might find answer to a physics question causal(one fact) which others might find to be multihop. I might mark a biology question multihop just because I don't know whether the information needed is a fact and can be present in a corpus.

- Out of the 1172 Questions one question appears to be of a comprehension type
  Qid: MCAS_2011_8_15365
  Question: Leopard's opening chant suggest that he is?
  A) happy B) confused C) confident D) generous

5.2.1 Result of the Analysis

Figure 5.1 shows distribution of reasoning types. Its clear from figure that the most common reasoning type are *Question Logic*, *Multihop Reasoning* and *Linguistic Matching* and are needed to be performed to answer 86.6, 84.13 and 91.55 percent of the questions respectively. *Causal Reasoning* is required only for 29.7% of the questions. These 29.7% questions too belong to other categories as well. There is only one questions which belongs to *Causal Reasoning* category alone. This indeed is the expected distribution because of the way ARC Challenge set is indirectly created by removing factoid questions(Refer 1.2.1).

It is important to note that the categorization is multi-label that is, questions usually belong to more than one category and the total number of questions is 1172 only. Most of the question depend on the options in some way or the other and since any question that require multihop reasoning will most likely require performing Linguistic matching, there are a large number

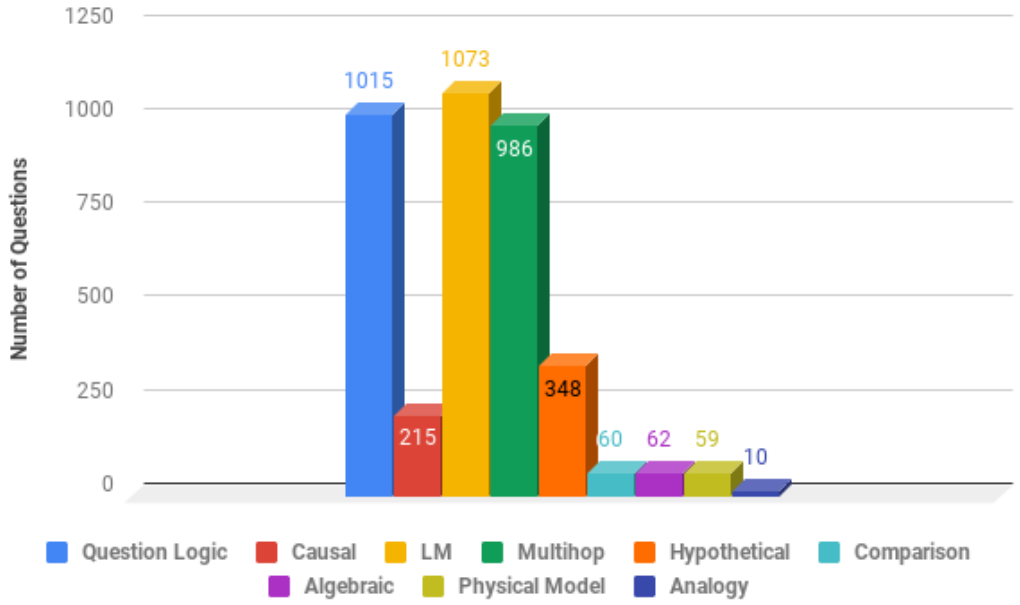

FIGURE 5.1: Distribution of reasoning types over questions

of common questions between Question Logic, Multihop and Linguistic Matching categories.

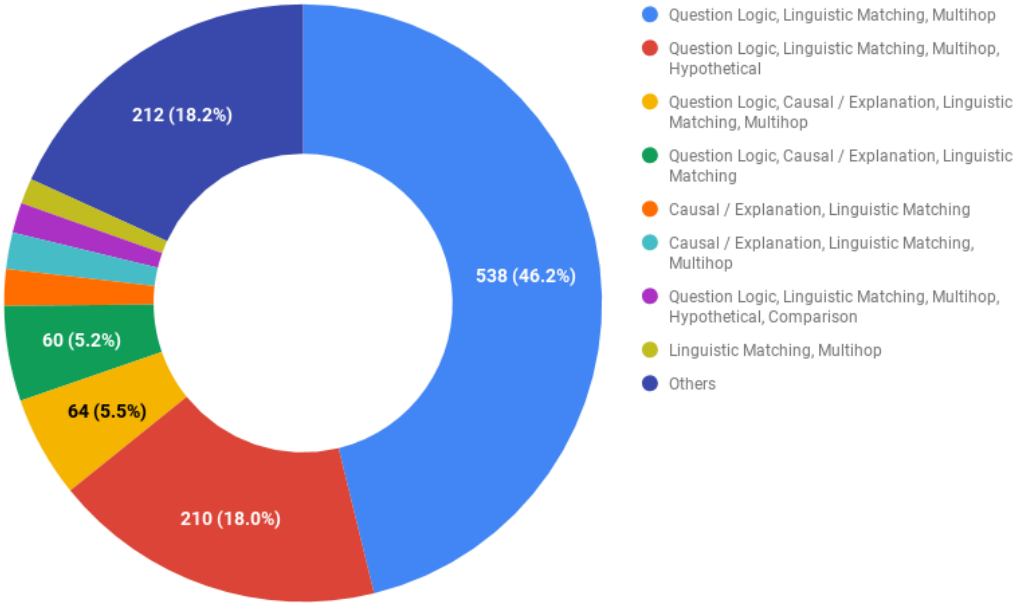

FIGURE 5.2: Distribution of questions over reasoning types

Others include 54 different combination of reasoning types. Each of them had less than or equal to 15 questions.

Another important thing to notice is that there are at least 62 questions which require some level of algebraic/numerical calculations to obtain the right

answer. For these 62 questions its highly unlikely that the correct answer to these question is present in the corpus in any form.

Figure 5.2 shows the other side of the coin. It shows the distribution of question over reasoning types. Almost half of the questions require all prominent three(*Question Logic*, *Multihop* and *Linguistic Matching*) types of reasoning in order to solve them. About 18% of require an additional ability to deal with hypothetical situation.

5.2.2 Performance Analysis based on Reasoning Types

Unfortunately apart from DGEM none of the models on the leader board could be used because none of them provide code or final trained models which can be used to generate their predictions. Only BiLSTM max-out [12] has provided the final trained model available but the code is uses old unsupported versions of libraries. So from the leader board only the DGEM model could be used for analysis.

We compare the performance of DGEM and our best scoring method that is Approach I(4.1) JC(4.1.4) model on the following three sets of questions.

- **Algebraic** Performance on all the 62 questions that require *Algebraic* reasoning. 5.1.
- **QL+MH+LM** Questions that require *Question Logic*, *Multihop* and *Linguistic Matching* type of reasoning but don't require *Algebraic* reasoning.
- **QL+CU+LM** Questions that require *Question Logic Causal* and *Linguistic Matching* type of reasoning but don't require *Algebraic* and *Multihop* reasoning.

Figure 5.3 compares the performance of the Approach I(JC) with the DGEM model on the algebraic type questions. As mentioned earlier, any QAS should not be able to answer these questions correctly. Although our approach gets almost same score as the DGEM method(21.48 and 21.77 respectively) a closer look at the graph shows that our model answered only 2 algebraic questions correctly while DGEM answered 9 out of which 3 definitely require solving proper equations.

Figure 5.4 and figure 5.5 compare the performance of the models on the questions in the *QL+MH+LM*(70% questions) and *QL+CU+LM* categories respectively. Although our model gets a better score overall in both the

categories however the number of correct questions it answered in the former category is about 18% less than that of DGEM. This indicates that our approach is missing multihop information from the corpus.

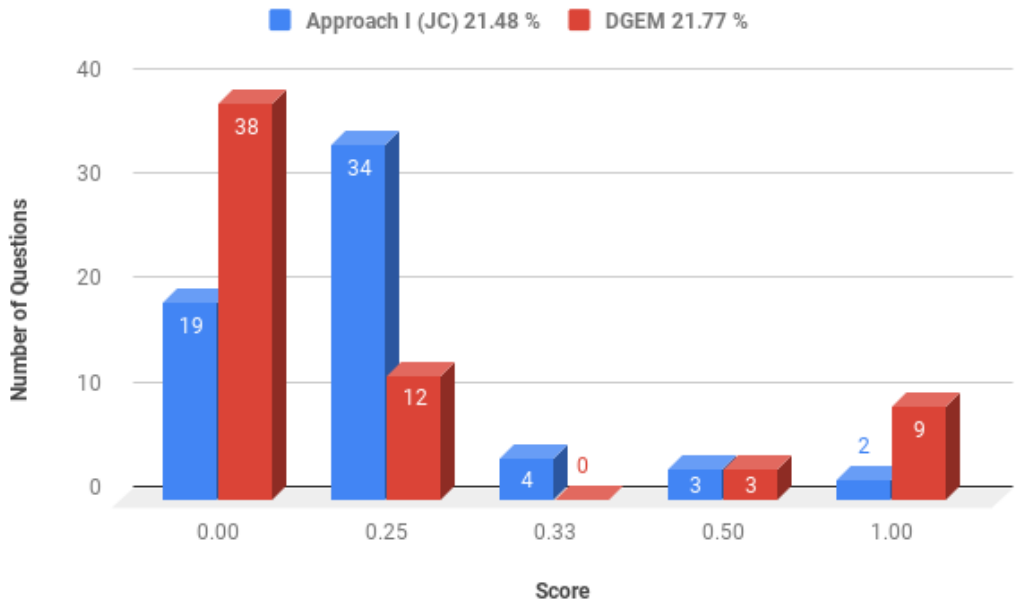

FIGURE 5.3: Performance comparison of Approach I(JC) and DGEM on Algebraic Questions

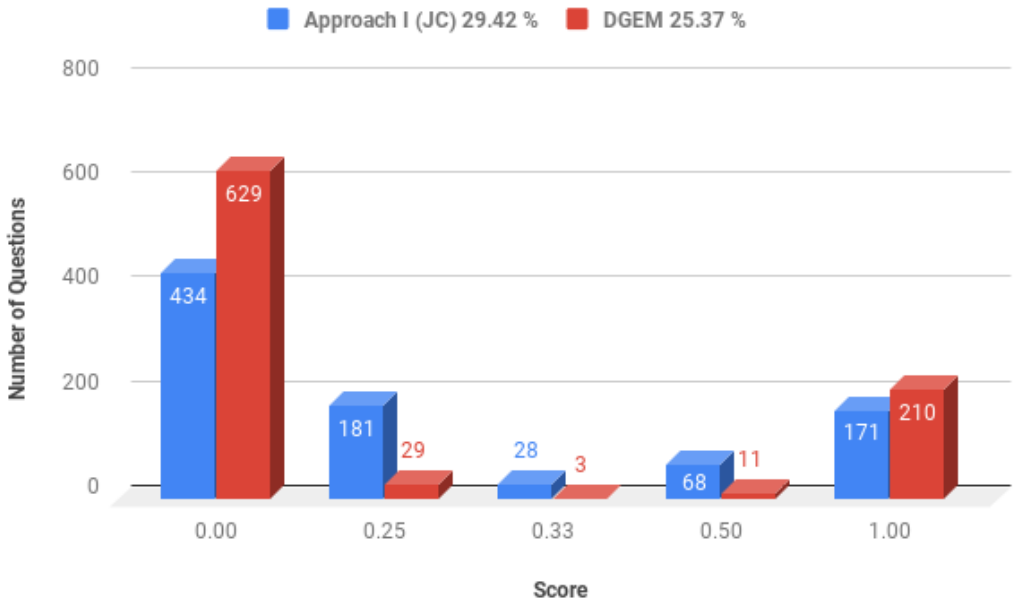

FIGURE 5.4: Performance comparison of Approach I(JC) and DGEM on *QL+MH+LM* Questions

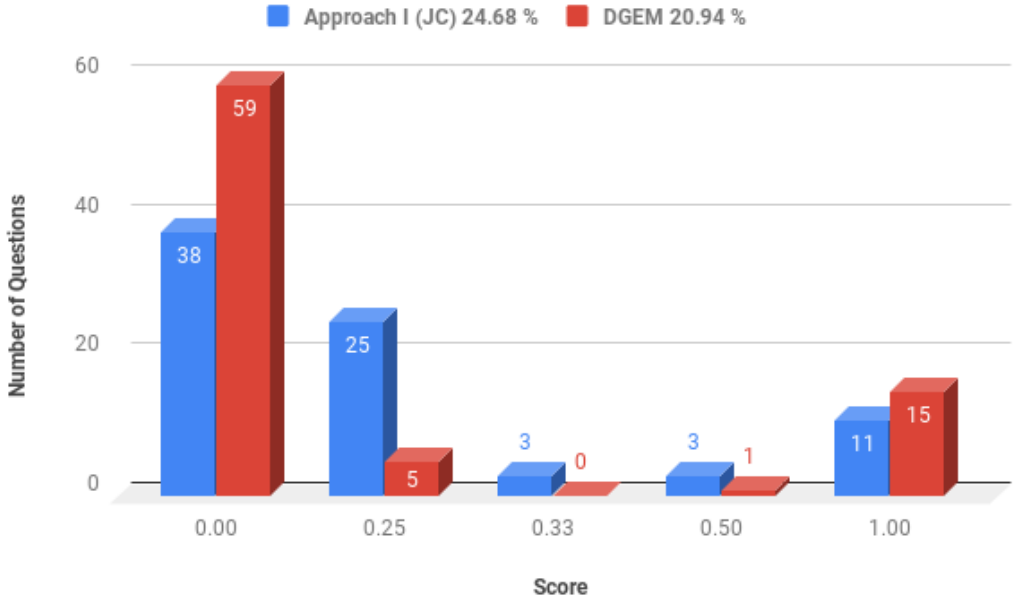

FIGURE 5.5: Performance comparison of Approach I(JC) and DGEM on *QL+MH+LM* Questions

Chapter 6

Conclusion and Future Work

The work done in this thesis can be broadly divided into two components. The first component covers the efforts and creation of non-neural model to solve the ARC Challenge. We successfully analyzed the problem and the then SOTA model, identified bottlenecks. Based on this analysis we proposed the *Approach I (JC)* 4.1, a graph based deterministic algorithm, which not only surpassed the highest scoring non-neural model but also gave about 2% better score the then SOTA DGEM model. **Note:** Neural based model with better scores than our model do exist now. We also proposed another graph based approach that although resulted in a better score than the DGEM model and a compact better graphical representation of the corpus graph but couldn't surpass the score set by the former approach. Under the second component we analyzed and successfully performed manual annotation of all the questions of the corpus with the types of reasoning required to answer these questions. The results of the annotation were in accordance with the rules that were used to filter out the questions and create the dataset. This strongly enforces the correctness of the annotations. By analyzing our best approach using the annotations gave us an insight that the approach is missing multihop information in the corpus.

This work is just the beginning as the challenge just started previous year and the field of Question Answering is one of the growing and hot field worldwide. This thesis shows that the there is still scope for non-neural methods. The performance of the approach I can be increased by focusing on integrating multihop information in its scoring algorithm. Approach II does incorporate multihop search but one needs to focus on reducing the scoring algorithms complexity so that a broader scope of the graph can be explored without blowing up the algorithm. The annotation performed on the question offers insight into the questions and the required reasoning. This insight can be utilized while designing the model which should improve performance.

For e.g. dividing the model in to ensemble of smaller models where each individual model is tweaked to answer the questions of a particular set of annotations. The annotation itself can be improved by getting the questions annotated by several people which would average out the bias problem.

Bibliography

[1] Peter Clark et al. “Think you have solved question answering? try arc, the ai2 reasoning challenge”. In: *arXiv preprint arXiv:1803.05457* (2018).

[2] Johannes Welbl, Nelson F Liu, and Matt Gardner. “Crowdsourcing multiple choice science questions”. In: *arXiv preprint arXiv:1707.06209* (2017).

[3] Guokun Lai et al. “RACE: Large-scale ReAding Comprehension Dataset From Examinations”. In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing* (2017). DOI: 10.18653/v1/d17-1082. URL: http://dx.doi.org/10.18653/v1/d17-1082.

[4] Pranav Rajpurkar et al. “SQuAD: 100,000+ Questions for Machine Comprehension of Text”. In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing* (2016). DOI: 10.18653/v1/d16-1264. URL: http://dx.doi.org/10.18653/v1/D16-1264.

[5] Peter Clark et al. “Combining retrieval, statistics, and inference to answer elementary science questions”. In: *Thirtieth AAAI Conference on Artificial Intelligence*. 2016.

[6] Ankur P. Parikh et al. “A Decomposable Attention Model for Natural Language Inference”. In: *EMNLP*. 2016.

[7] Min Joon Seo et al. “Bidirectional Attention Flow for Machine Comprehension”. In: *CoRR* abs/1611.01603 (2017).

[8] Tushar Khot, Ashish Sabharwal, and Peter Clark. “SciTaiL: A Textual Entailment Dataset from Science Question Answering”. In: *AAAI*. 2018.

[9] Yuyu Zhang et al. “KG2: Learning to Reason Science Exam Questions with Contextual Knowledge Graph Embeddings”. In: *CoRR* abs/1805.12393 (2018).

[10] Wanjun Zhong et al. “Improving Question Answering by Commonsense-Based Pre-Training”. In: *CoRR* abs/1809.03568 (2018).

[11] Robyn Speer, Joshua Chin, and Catherine Havasi. *ConceptNet 5.5: An Open Multilingual Graph of General Knowledge*. 2016. arXiv: 1612.03975 [cs.CL].

[12] Alexis Conneau et al. "Supervised Learning of Universal Sentence Representations from Natural Language Inference Data". In: *EMNLP*. 2017.

[13] Kai Sun et al. *Improving Machine Reading Comprehension with General Reading Strategies*. 2018. arXiv: 1810.13441 [cs.CL].

[14] Alec Radford et al. "Improving language understanding by generative pre-training". In: *URL https://s3-us-west-2. amazonaws. com/openai-assets/research-covers/languageunsupervised/language understanding paper. pdf* (2018).

[15] Jacob Devlin et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2018. arXiv: 1810.04805 [cs.CL].

[16] Microsoft Dynamics 365 AI Research. *BERT MRC Transfer(Single Model)*. 2018. URL: https://leaderboard.allenai.org/arc/submission/bgaj76p23n20bdi7i2ng.

[17] Dian Yu (Tencent AI Lab) Kai Sun (Cornell/Tencent AI Lab). *QA Transfer*. 2018. URL: https://leaderboard.allenai.org/arc/submission/bgkqp1123n26brb8hl2g.

[18] Microsoft Dynamics 365 AI Research. *MultiTask BERT (Single Model)*. 2019. URL: https://leaderboard.allenai.org/arc/submission/bise0anl77gjp9grm6k0.

[19] Aristo team at Allen Institute for AI. *BERT-large "whole word masking" followed by several steps of fine-tuning, providing each question with a context of sentences retrieved from a large corpus.* 2019. URL: https://leaderboard.allenai.org/arc/submission/bk5snmbvhqhm94h7heag.

[20] Daniel Khashabi et al. "Question Answering via Integer Programming over Semi-Structured Knowledge". In: *IJCAI*. 2016.

[21] Niket Tandon, Gerard de Melo, and Gerhard Weikum. "Webchild 2.0: Fine-grained commonsense knowledge distillation". In: *Proceedings of ACL 2017, System Demonstrations* (2017), pp. 115–120.

[22] *NCERT Textbooks*. URL: http://ncert.nic.in/textbook/textbook.htm.

[23] Christopher D. Manning et al. "The Stanford CoreNLP Natural Language Processing Toolkit". In: *Association for Computational Linguistics (ACL) System Demonstrations*. 2014, pp. 55–60. URL: http://www.aclweb.org/anthology/P/P14/P14-5010.

[24] Radityo Eko Prasojo, Mouna Kacimi, and Werner Nutt. "StuffIE: Semantic Tagging of Unlabeled Facets Using Fine-Grained Information Extraction". In: *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. ACM. 2018, pp. 467–476.

[25] Michael Boratko et al. *A Systematic Classification of Knowledge, Reasoning, and Context within the ARC Dataset*. 2018. arXiv: 1806.00358 [cs.AI].