



IIT DELHI

MASTERS THESIS

Question Answering: AI2 Reasoning Challenge

Author:

Saurabh Godse

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 Retrieval Group
Department of Computer Science and Engineering

July 12, 2020

CERTIFICATE

This is to certify that the thesis titled **Question Answering: AI2 Reasoning Challenge**, being submitted by **Saurabh Godse**, 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

July 12, 2020

Abstract

Question Answering: AI2 Reasoning Challenge

by Saurabh Godse

Question Answering is one of the trendings and growing research field worldwide. *QuaRel* is a new and exciting challenge (November 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 physics MCQs.

The work done in this thesis can broadly be divided into two parts. The first part is about information extraction in which we gather all information needed to train the model and to build the knowledge graphs. In this part, we acquired concepts in physics e.g. velocity, friction from Wikipedia pages that are tagged physics and the relationship among them called QuaRel relationships. To find these relationships we used the mathematical formulas. By parsing formula direct and inverse relations among entities can be obtained.

The second part's goal is given a natural language question we would like to identify which physics concepts are required to answer it. For this firstly we'll tag the sentence using POS-TAG. Then we've created chunks using Regex. And these chunks are further processed. In this to approaches were used Wordnet and Sci-Bert.

Acknowledgements

First and foremost, I would like to thank my mentor Dr. Maya Ramanath for permitting me to work under her supervision. Without her practice and supervision, this work would not have been possible. I would additionally like to thank my pals and family for their steady moral support and motivation.

I would like to express my gratitude in the direction of the helpful people online (StackOverflow, Reddit, etc.) This work would've moved appreciably slowly had it not been for them

Contents

Abstract	ii
Acknowledgements	iii
1 Introduction	1
1.1 Question Answering	1
1.1.1 Types of QA Systems	2
1.2 QuaRel	4
1.2.1 Knowledge Representation	4
1.2.2 Dataset	6
1.2.3 Performance Evaluation Criteria	7
2 Literature Survey	8
3 The 3 Stages	10
4 Background Knowledge	12
4.1 Acquire Knowledge	12
4.1.1 Stanford POS tagger	12
4.1.2 Named Entity Recognizer	12
4.1.3 Using 'title' of page as entity	13
4.2 Relationship between concepts	14
5 Mapping Question to Logical Form	18
5.1 Concepts related to Question	18
5.1.1 Wordnet	18
5.1.2 The Approach	18
5.1.3 Drawbacks in Wordnet results	20
5.2 Word2Vec	22
5.2.1 Word embeddings	22
5.2.2 Gensim	22
5.2.3 Spacy	23
5.2.4 Cosine similarity	23

5.2.5	Observations	24
5.2.6	BERT	25
5.2.7	The Approach using BERT	25
5.2.8	Sci-Bert	26
5.2.9	The approach using Sci-BERT	27
5.3	Results	28
6	Conclusion and Future Work	31
	Bibliography	32

List of Figures

1.1	Qualitative Story Problem	5
3.1	3 stages of project	10
4.1	Snapshot of results of Stanford POS tagger	13
4.2	Snapshot of results of Named Entity Recognizer	14
4.3	Snapshot of Petscan	14
4.4	Result of petscan	15
4.5	List of equations in Kinematics	15
4.6	Map of Classical mechanics physics concepts	16
4.7	Classical Mechanics equations after LaTeX to English conversion	17
4.8	Relationships between classical mechanics concepts	17
5.1	Result of searching 'push' on wordnet	19
5.2	Sample question	20
5.3	Chunks of sample question	20
5.4	Results for sample question	20
5.5	Drawbacks of Wordnet	21
5.6	Results of searching word 'LESS' on Wordnet	21
5.7	Result of cosine-similarities between words given by Gensim .	24
5.8	Result of cosine-similarities between words given by Spacy . .	24
5.9	Result of cosine-similarities between clause and words given by Spacy	24
5.10	Result of Spacy for sample question	24
5.11	Result of BERT for sample question	26
5.12	Result of Sci-BERT for sample question	27
5.13	First heatmap	28
5.14	Second heatmap	29
5.15	Third heatmap	30
6.1	Small portion of Knowledge Graph	31

List of Tables

1.1	Division of QuaRel Questions in each category	7
2.1	Scores of different models on full QUAREL dataset and <i>QUAREL^F</i> subset about friction [5]	9
5.1	Results achieved in various approaches	30

Chapter 1

Introduction

1.1 Question Answering

In this era of information, as the users find it tough to navigate a large amount of wealth of online information, the automated question answering system's necessity becomes more urgent. A system to which users can ask a natural language question and it'll respond quickly and accurately with a valid answer having sufficient context. Question Answering systems or QASs address this problem.

The computer science discipline which focuses on building systems that can answer natural language questions posed by humans is called Question answering. For a system/computer to understand natural language consists of the program system capable of translating sentences into internal representation so that this system can give a valid answer to the questions asked by users. By saying *valid answer* here, it means answers relevant to question asked by the user.

It encompasses numerous other computer science fields such as **Natural Language Processing(NLP)**, **Information Retrieval(IR)** and **Information Extraction(IE)**. 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 based on what is answered.

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**

The application domain is categorized into two categories Restricted area and Open-domain QASs from which the questions are requested by the users. The repository of questions is constrained in a constrained domain, unlike open-domain questions. Different techniques are required to reply to constrained area questions which be counted on area unique ontology and terminology in contrast to open area questions that remember well-known ontology and world know-how to get an ultimate answer. Google search, for example, is an open area system the place where a consumer can search for any information. A College Library's search engine, on the other hand, is a limited 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**

Introduction of answers to the users in quite a several forms that can be extracted as textual content snippet taken from supply files or generated answers. The form of solutions generally depends upon users' questions. Generally, the factoid or list questions have answers in the structure of sentences. Causal, hypothetical questions have answers in the form of passages. Confirmation questions have generated solutions in the form of both yes or No. Some Opinionated questions have answers in 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 dialog as the answer. For example, CoQA is a large-scale dataset for building Conversational Question Answering Systems. The purpose of the CoQA challenge is to measure the ability of machines to understand a text passage and reply to a series of interconnected questions that appear in a conversation.

In this thesis we'll deal with the QuaRel or AI2 reasoning challenge which is restricted to the domain of Grade school level physics MCQs. Details of the QuaRel are explained in the next section.

1.2 QuaRel

AI2 reasoning challenge (ARC) is a question answering task proposed by Clark et al. [5] In this challenge the aim is to create a Question Answering system that can answer the questions in the QuaRel Dataset.

1.2.1 Knowledge Representation

Let's look at framework for representing questions first and the knowledge to answer them. The dataset, described later, includes logical forms expressed in this language. We use the same notations used in the paper [5].

- **Qualitative Background Knowledge**

A simple representation of qualitative relationships is used, leveraging prior work in qualitative reasoning. Let $P = p_i$ be the set of properties relevant to the question set's domain (e.g., smoothness, friction, speed). Let $V_i = v_{ij}$ be a set of qualitative values for property p_i (e.g., fast, slow). For the history information about the domain itself (a qualitative model), following [5], the following predicates are used:

1. $q+(property1, property2)$
2. $q-(property1, property2)$

$q+$ denotes that property1 and property2 are qualitatively proportional, e.g., if property1 goes up, property2 will too, while $q-$ denotes inverse proportionality, e.g.,

If friction goes up, speed goes down. $q-(friction, speed)$. We also introduce the predicate:

- $higher-than(val_{ij}, val_{ik}, property_i)$

where $val_{ij} \in V_i$, allowing an ordering of property values to be specified, e.g., $higher-than(fast, slow, speed)$. For simplification purposes here, just two property values are used, low and high, for all properties.

- **Representing Questions**

One of the main features in our representation is creation of concepts of questions when describing events occurring in two separate worlds being compared. Such comparison can also be drawn between a world at time $t1$ and $t2$. E.g., in Figure 1.1 the two worlds being compared are the car on wood, and the car on carpet. The tags world1 and world2

denote these different situations, and semantic parsing requires learning to correctly associate these tags with parts of the question describing those situations. This abstracts away irrelevant details of the worlds, while still keeping track of which world is which.

Two predicates to express qualitative information in questions is defined below :

- $qrel(\text{property}, \text{direction}, \text{world})$
- $qval(\text{property}, \text{value}, \text{world})$

where $\text{property } (p_i) \in P$, $\text{value} \in V_i$, $\text{direction} \in \text{higher}, \text{lower}$, and $\text{world} \in \text{world1}, \text{world2}$. $qrel()$ denotes the relative assertion that property is higher/lower in world compared with the other world, which is left implicit e.g. The car rolls further on wood.

$qrel(\text{distance}, \text{higher}, \text{world1})$

where world1 is a tag for the “car on wood” situation (hence world2 becomes a tag for the opposite “car on carpet” situation). $qval()$ denotes that property has an absolute value in world, e.g. The car’s speed is slow on carpet.

$qval(\text{speed}, \text{low}, \text{world2})$

Qualitative Story Problem:

Alan noticed that his toy car rolls further on a wood floor than on a thick carpet. This suggests that:

- (A) The carpet has more resistance
- (B) The floor has more resistance

Solution: (A) The carpet has more resistance

Identification of worlds being compared:



Question Interpretation (Logical Form):

$qrel(\text{distance}, \text{higher}, \text{world1}) \rightarrow$
 $qrel(\text{friction}, \text{higher}, \text{world2}) ;$
 $qrel(\text{friction}, \text{higher}, \text{world1})?$

FIGURE 1.1: Qualitative Story Problem

- **Logical Forms for Questions**

The space of logical forms (LFs) for the questions that is considered is relatively compact despite the wide variation in language. For every question, the body of question creates a scenario and each answer option then probes an implication. Thus a question's LF is expressed as a tuple: (setup, answer-A, answer-B)

where setup is the predicate(s) describing the scenario, and answer-* are the predicate(s) being queried for.

1.2.2 Dataset

QUAREL dataset contains 2771 multiple-choice questions, with their logical forms. Logical forms are used for semantic parsing. This dataset contains several such annotated logical forms. It's size is similar to several other datasets with such logical forms.

The multiple choice questions in the dataset are designed in such a way that multiple people can look at it in different ways. People can be imaginative and can define question in terms of their own worlds.

The novel technique of reverse-engineering is used to elicit the LFs from set of follow-up questions. To move towards the answer of question one should try to find the answer of following questions :

1. What is the correct answer (A or B)?
2. Which property are the answer options asking about?
3. In the correct answer, is this property higher or lower than in the incorrect answer?
4. Do the answer options:
 - ask the same question about different objects/situations?
 - ask opposite questions about the same object/situation?
5. Which direction of comparison is used in the body of the question?
 - higher/lower?
 - OR were two values given? If so, enter the values, standardized as high/low in the LF?

Corpus

The corpus used in first step for gathering information is Wikipedia pages that are tagged physics. There are around 14M science related articles from which we can extract physics concepts and formulas to get the relationship among concepts. The sources include science-related documents from net, Wiktionary, articles from Wikipedia that were tagged science etc.

1.2.3 Performance Evaluation Criteria

Dataset is already divided into Train, Dev and Test sets.

	Total Questions
Train	1941
Dev	278
Test	552
Total	2771

TABLE 1.1: Division of QuaRel Questions in each category

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.

Four systems are used to evaluate the difficulty of this dataset.

1. Information Retrieval (IR) System

Given a question q , this system searches it in a huge (280GB) text corpus. If question q with an answer option is loosely found in the corpus this system returns the confidence. To do this, for each answer option a_i , it sends $q + a_i$ as a query to a search engine and returns the search engine's score for the top retrieved sentences where it also has at least one non-stopword overlap with q , and at least one with a_i . The option with the highest score is selected[5].

2. Pointwise Mutual Information (PMI)

The co-occurring words may provide some information for answering these questions, e.g., the high co-occurrence of "faster" and "ice" in a corpus may help answer a question ending with "...faster? (A) ice (B) gravel". To formalize this, given a question q and an answer option a_i , we use PMI to measure the strength of the associations between parts of q and parts of a_i [5].

3. Rule-based Semantic Parser

In rule-based semantic parser first, the question is pre-processed to tag likely references to the worlds being compared, using hand-written rules that look for surface names ("road", "ice"), appropriate adjectives ("rough", "green"), and by position ("over <X>"). The first candidate word/phrase is tagged world1 (with type WORLD), the second world2, and if those phrases occur later in the question, they are tagged with the corresponding world. The system then parses the question using 142

task-specific, CCG-like rules, such as:

$$”isgreaterthan” \rightarrow (SPROPERTY)WORLD$$

$$”velocity” \rightarrow PROPERTY : speed$$

where

WORLD means “look left for something of category *WORLD*”. Thus a tagged phrase like “the velocity on ice[word2] is greater than” produces `qrel(speed, higher, world2)`. The parser skips over most words in the story, only paying attention to words that are tagged or referenced in the grammar.[5]

4. Type-constrained Neural Semantic Parser (QUASP)

This system is AllenNLP’s implementation of a neural semantic parser (Gardner et al. 2018). This parser uses a type-constrained, encoder-decoder architecture, representative of the current state-of-the-art on many datasets. The model architecture is similar to standard seq2seq models, with an LSTM that encodes the question and an LSTM with attention over the encoded question that decodes a logical form.

The above systems are ran over QUAREL dataset and the results are shown in the following tables:

Dataset→ Model↓	QUAREL		QUAREL ^F	
	Dev	Test	Dev	Test
Random	50.0	50.0	50.0	50.0
Human	96.4	-	95.0	-
IR	50.7	48.6	50.7	48.9
PMI	49.3	50.5	50.7	52.5
Rule-Based	-	-	55.0	57.7
BILSTM	55.8	53.1	59.3	54.3
QUASP	62.1	56.1	69.2	61.7
QUASP+	68.9	68.7	79.6	74.5

TABLE 2.1: Scores of different models on full QUAREL dataset and *QUAREL^F* subset about friction [5]

Chapter 3

The 3 Stages

The project is broadly divided into 3 stages :

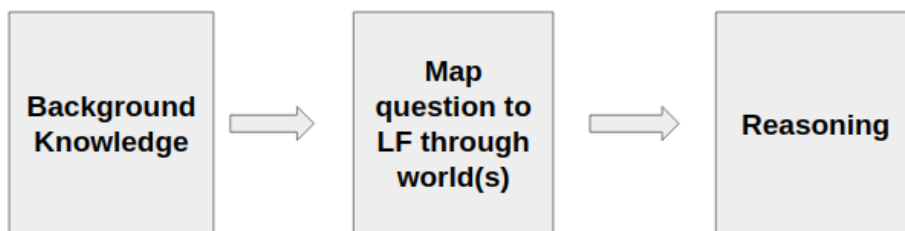


FIGURE 3.1: 3 stages of project

1. Background Knowledge

The first stage of project focuses on gathering data using which we can build a quality knowledge graph in further stages which will be used for reasoning by model. This stage aims on two things : *Acquire knowledge* and *Relationship between concepts*. Acquiring knowledge means you are supposed to gather the physics concepts to find the relationship between them. Relationship between concepts can be of two types *direct* and *inverse* denoted by $q+$ and $q-$ respectively.

2. Map question to LF through world(s)

The second stage of project concentrates on mapping the question to it's Logical Forms (LF). For this firstly we need to find the physics concepts related to question and hence the ultimate goal becomes "*Given a natural language question, we would like to identify what physics concepts are required to answer it*". After completing this, we've to find mapping of question to various logical forms by using the related physics concepts.

3. Reasoning

The third stage of project is about building knowledge graph from the QUAREL relations that we gathered in first stage and then develop algorithms capable of doing precise reasoning by looking at knowledge graph and logical forms of questions.

Chapter 4

Background Knowledge

4.1 Acquire Knowledge

As mentioned above this is first stage of project. The stage focuses on gathering data which will form strong foundation for further stages. In this part we've parsed around 40k Wikipedia pages tagged *physics* to find the physics concepts. For this we've used 3 approaches :

4.1.1 Stanford POS tagger

Tagging is a kind of classification that may be defined as the automatic assignment of description to the tokens. Here the descriptor is called tag, which may represent one of the part-of-speech, semantic information and so on [6].

POS tagging is Part Of Speech tagging which assigns tags as parts of speech to words.

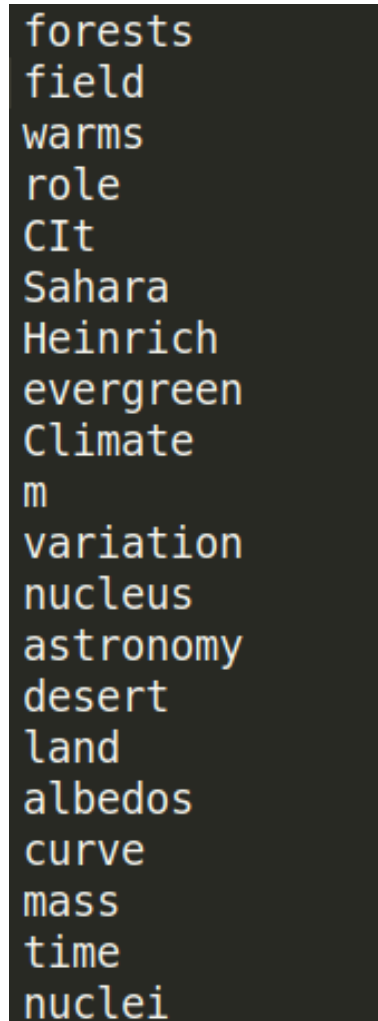
After using Stanford POS tagger we've came to conclusion that this tagger is giving lot of useless entities. The figure 4.1 is snapshot of the results of Stanford POS tagger.

4.1.2 Named Entity Recognizer

The process of finding names, people, places, and other entities, from a given text is known as Named Entity Recognition (NER)[7].

To perform various NER tasks, OpenNLP uses different predefined models namely, en-nerdate.bn, en-ner-location.bin, en-ner-organization.bin, en-ner-person.bin, and en-ner-time.bin. All these files are predefined models which are trained to detect the respective entities in a given raw text.

The `opennlp.tools.namefind` package contains the classes and interfaces that are used to perform the NER task. To perform NER task using OpenNLP library, you need to :



```
forests
field
warms
role
CIt
Sahara
Heinrich
evergreen
Climate
m
variation
nucleus
astronomy
desert
land
albedos
curve
mass
time
nuclei
```

FIGURE 4.1: Snapshot of results of Stanford POS tagger

- Load the respective model using the **TokenNameFinderModel** class.
- Instantiate the **NameFinder** class.
- Find the names and print them.

The problem with this approach was that NER was not tagging the physics concepts. It just tags the parts of speech in the sentence. The figure 4.2 shows results of NER.

4.1.3 Using 'title' of page as entity

As stanford POS tagger and Named Entity Recognizer failed to give us what we want we decided to treat title of Wikipedia page as concept and try to gather titles of the pages related to them up to 3-4 depth. For this we've used pet scan of Wikipedia which gives facility to enter the title and it'll give you

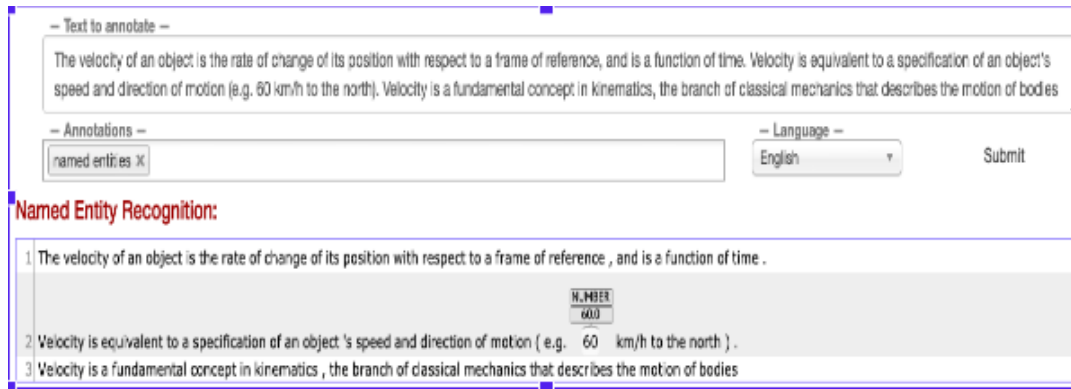


FIGURE 4.2: Snapshot of results of Named Entity Recognizer

related titles in the way you wanted (i.e. list of wiki page titles or list of wiki page titles with their categories). The figure 4.3 shows the petscan interface and figure 4.4 shows the result given by petscan.

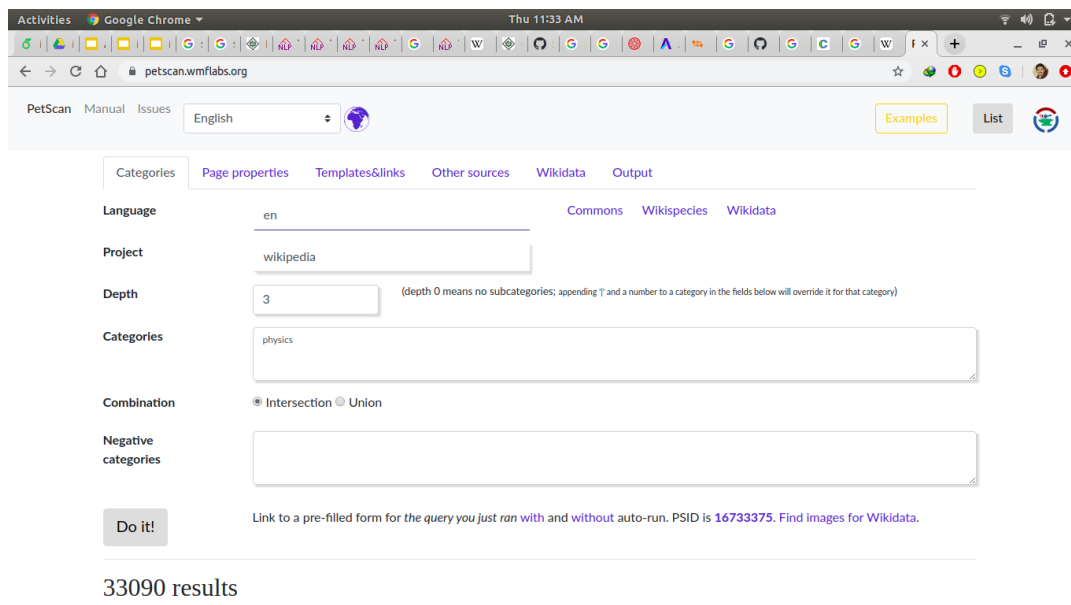
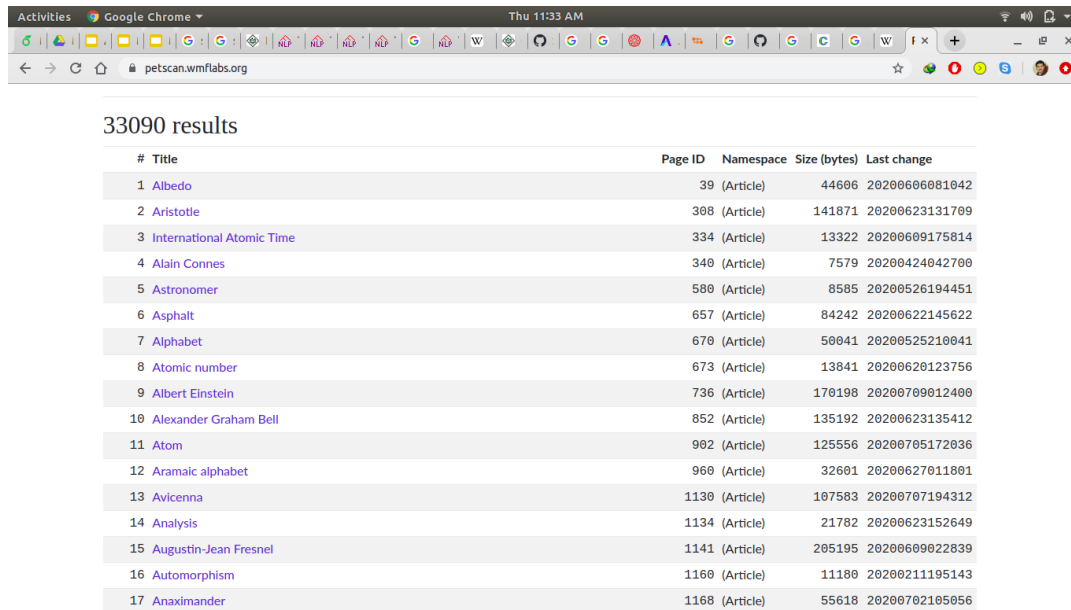


FIGURE 4.3: Snapshot of Petscan

4.2 Relationship between concepts

Now as we have list of physics concepts, our next step becomes to find the relationship between them. For this we've used two python libraries *BeautifulSoup* and *Sympy*. By observing the Wikipedia pages we've come to conclusion that from the tables in Wikipedia pages we can extract mathematical formulas and from those formulas we can get the direct or inverse relationship between physics entities. Also from those tables we can



33090 results

#	Title	Page ID	Namespace	Size (bytes)	Last change
1	Albedo	39	(Article)	44606	20200606081042
2	Aristotle	308	(Article)	141871	20200623131709
3	International Atomic Time	334	(Article)	13322	20200609175814
4	Alain Connes	340	(Article)	7579	20200424042700
5	Astronomer	580	(Article)	8585	20200526194451
6	Asphalt	657	(Article)	84242	20200622145622
7	Alphabet	670	(Article)	50041	20200525210041
8	Atomic number	673	(Article)	13841	20200620123756
9	Albert Einstein	736	(Article)	170198	20200709012400
10	Alexander Graham Bell	852	(Article)	135192	20200623135412
11	Atom	902	(Article)	125556	20200705172036
12	Aramaic alphabet	960	(Article)	32601	20200627011801
13	Avicenna	1130	(Article)	107583	20200707194312
14	Analysis	1134	(Article)	21782	20200623152649
15	Augustin-Jean Fresnel	1141	(Article)	205195	20200609022839
16	Automorphism	1160	(Article)	11180	20200211195143
17	Anaximander	1168	(Article)	55618	20200702105056

FIGURE 4.4: Result of petscan

build map which maps physics concept to its symbol e.g., velocity : v and vice versa. The figure 4.5 shows table on wikipedia page about classical mechanics.

Quantity (common name/s)	(Common) symbol/s	Defining equation	SI units	Dimension
Velocity	\mathbf{v}	$\mathbf{v} = d\mathbf{r}/dt$	m s^{-1}	$[\text{L}][\text{T}]^{-1}$
Acceleration	\mathbf{a}	$\mathbf{a} = d\mathbf{v}/dt = d^2\mathbf{r}/dt^2$	m s^{-2}	$[\text{L}][\text{T}]^{-2}$
Jerk	\mathbf{j}	$\mathbf{j} = d\mathbf{a}/dt = d^3\mathbf{r}/dt^3$	m s^{-3}	$[\text{L}][\text{T}]^{-3}$
Jounce	\mathbf{s}	$\mathbf{s} = d\mathbf{j}/dt = d^4\mathbf{r}/dt^4$	m s^{-4}	$[\text{L}][\text{T}]^{-4}$
Angular velocity	$\boldsymbol{\omega}$	$\boldsymbol{\omega} = \hat{\mathbf{n}} (d\theta/dt)$	rad s^{-1}	$[\text{T}]^{-1}$
Angular Acceleration	$\boldsymbol{\alpha}$	$\boldsymbol{\alpha} = d\boldsymbol{\omega}/dt = \hat{\mathbf{n}} (d^2\theta/dt^2)$	rad s^{-2}	$[\text{T}]^{-2}$
Angular jerk	$\boldsymbol{\zeta}$	$\boldsymbol{\zeta} = d\boldsymbol{\alpha}/dt = \hat{\mathbf{n}} (d^3\theta/dt^3)$	rad s^{-3}	$[\text{T}]^{-3}$

FIGURE 4.5: List of equations in Kinematics

- **BeautifulSoup** BeautifulSoup is an XML/HTML parser for Python that can turn even invalid markup into a parse tree. It provides simple, idiomatic ways of navigating, searching, and modifying the parse tree. Given a Wikipedia page URL we can extract the tables on that page using BeautifulSoup[8].
- **Sympy** SymPy is a Python library for symbolic mathematics. It aims to become a full-featured computer algebra system (CAS) while keeping the code as simple as possible in order to be comprehensible and easily extensible. SymPy is written entirely in Python[9].

Using BeautifulSoup we've extracted tables on Wikipedia pages. Wikipedia stores mathematical formulas in LaTeX form. To parse those formulae we've used Sympy. From the information in tables map is built first. Then parser is used to convert latex to English. The relationship between the entities in formula can be determined by mathematical operators(+, -, *, /). The figure 4.6 shows the map built after parsing table, the figure 4.7 shows the equations we got after converting LaTeX formulas to English and figure 4.8 Quarel relationship between classical mechanics concepts that we derived from the formulas.

```
Quantity (common name/s),(Common) symbol/s
Velocity,v
Acceleration,a
Jerk,j
Jounce,s
Angular velocity,omega
Angular Acceleration,alpha
Angular jerk,zeta
Momentum,p
Force,F
Impulse,"J, Deltap, I"
Angular momentum,"L, S"
Torque,tau
Angular impulse,DeltaL
Work,W
Energy,"U, E"
Power,P
Mass,m
Time,t
Distance,r
Angular distance,theta
```

FIGURE 4.6: Map of Classical mechanics physics concepts


```
Eq(v, ((d*(r)))/((d*t)))
Eq(Eq(a, ((d*(v)))/((d*t))), ((d**2*(r)))/((d*t**2)))
Eq(Eq(j, ((d*(a)))/((d*t))), ((d**3*(r)))/((d*t**3)))
Eq(Eq(s, ((d*(j)))/((d*t))), ((d**4*(r)))/((d*t**4)))
```

FIGURE 4.7: Classical Mechanics equations after LaTeX to English conversion

```
# -- classical mechanics
q-( Force , Time )
q+( Angular momentum , Momentum )
q+( Velocity , Distance )
q+( Angular impulse , Distance )
q-( Jerk , Time )
q+( Momentum , Mass )
q+( Power , Momentum )
q+( Angular Acceleration , Angular distance )
q+( Momentum , Distance )
q+( Velocity , Acceleration )
```

FIGURE 4.8: Relationships between classical mechanics concepts

Chapter 5

Mapping Question to Logical Form

This task can be divided into 2 sub-tasks. First is given a natural language question identify which physics concepts are required to answer it and second is to find mappings of those concepts.

5.1 Concepts related to Question

To find the concepts related to given natural language question, we've used two approaches. The first one uses Wordnet library of python and second is to use word embeddings i.e. Word2Vec.

5.1.1 Wordnet

WordNet is a large lexical database of English. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept. Synsets are interlinked by means of conceptual-semantic and lexical relations[10].

WordNet groups words together based on their meanings. However, there are some important distinctions. First, WordNet interlinks not only word forms (strings of letters) but also specific senses of words. As a result, words that are close to one another by meaning are disambiguated in the network. Second, WordNet labels the semantic relations among words, whereas the groupings of words does not follow any explicit pattern other than meaning similarity. The figure 5.1 shows the results of Wordnet after searching word *push*.

5.1.2 The Approach

Following points are step by step approach that we've used to identify the concepts related to question using *Wordnet* :

WordNet Search - 3.1

- [WordNet home page](#) - [Glossary](#) - [Help](#)

Word to search for:

Display Options:

Key: "S:" = Show Synset (semantic) relations, "W:" = Show Word (lexical) relations

Display options for sense: (gloss) "an example sentence"

Noun

- **S: (n) push, pushing** (the act of applying **force** in order to move something away) "he gave the door a hard push"; "the pushing is good exercise"
- **S: (n) push, thrust** (the **force** used in pushing) "the push of the water on the walls of the tank"; "the thrust of the jet engines"
- **S: (n) energy, push, get-up-and-go** (enterprising or ambitious drive) "Europeans often laugh at American energy"
- **S: (n) push button, push, button** (an electrical switch operated by pressing) "the elevator was operated by push buttons"; "the push beside the bed operated a buzzer at the desk"
- **S: (n) push** (an effort to advance) "the army made a push toward the sea"

Verb

- **S: (v) push, force** (move with **force**) "He pushed the table into a corner"
- **S: (v) push, bear on** (press, drive, or impel (someone) to action or completion of an action) "He pushed her to finish her doctorate"
- **S: (v) advertise, advertize, promote, push** (make publicity for; try to sell (a product)) "The salesman is aggressively pushing the new computer model"; "The company is heavily advertizing their new laptops"

FIGURE 5.1: Result of searching 'push' on wordnet

- *Glossary of physics* : It is the collection of physics concepts we've gathered in step 1.
- Given a natural language question firstly we are tagging the question using POS-TAG(Parts of Speech Tagger).
- Create chunks from question based on RegEx. e.g. <VB> <JJ> <IN> <DT> <NN>. The figure 5.2 shows sample question and figure 5.3 shows chunks of that question we got.
- Then we have extracted verbs and adjectives from those chunks. The reason to extract only verbs and adjectives is because from them only we can get the physics concepts related to the question.
- Now we give this verbs and adjectives to Wordnet which gives us synonyms set and definitions of that word.

- Tokenization and lemmitization of synonyms set which we got from wordnet is done.
- The words are matched with glossary and the matched words are called as concepts related to the question. The figure 5.4 contains the result of sample question.

```
Question -->
The propeller on Kate's boat moved slower in the ocean compared to the river.
This means the propeller heated up less in the ____
(A) ocean (B) river
```

FIGURE 5.2: Sample question

```
Chunks -->
(NP moved/VBD slower/JJR in/IN the/DT ocean/NN)
(NP compared/VBN to/TO the/DT river/NN)
(NP means/VBZ the/DT propeller/NN)
(NP heated/VBD up/RP less/RBR in/IN the/DT (A)/JJ ocean/JJ)
```

FIGURE 5.3: Chunks of sample question

```
Results -->
{'heat', 'mass', 'work', 'volume', 'motion'}
```

FIGURE 5.4: Results for sample question

5.1.3 Drawbacks in Wordnet results

- By looking at the results for the question, we can say that Wordnet is giving us desired results but along with them it is having some impurities.
- Impurity means it is having some words which are not supposed to be in the results.
- Observe the figure 5.5, it is showing that from word *MOVED* we've got *MOTION*, from word *HEATED* we've got *HEAT* but from word *LESS* we've got *MASS*.

```

moved
move.v.03
move so as to change position, perform a nontranslational motion
The word found in dictionary --> motion

heated
heat.v.01
make hot or hotter
The word found in dictionary --> heat

less
less.a.01
(comparative of 'little' usually used with mass nouns)
The word found in dictionary --> mass

```

FIGURE 5.5: Drawbacks of Wordnet

WordNet Search - 3.1
[- WordNet home page](#) - [Glossary](#) - [Help](#)

Word to search for:

Display Options:

Key: "S:" = Show Synset (semantic) relations, "W:" = Show Word (lexical) relations
 Display options for sense: (gloss) "an example sentence"

Adjective

- S: (adj) less** ((comparative of 'little' usually used with **mass** nouns) a quantifier meaning not as great in amount or degree) *"of less importance"; "less time to spend with the family"; "a shower uses less water"; "less than three years old"*
- S: (adj) less** ((usually preceded by 'no') lower in quality) *"no less than perfect"*
- S: (adj) less** ((nonstandard in some uses but often idiomatic with measure phrases) fewer) *"less than three weeks"; "no less than 50 people attended"; "in 25 words or less"*
- S: (adj) small, little** (limited or below average in number or quantity or magnitude or extent) *"a little dining room"; "a little house"; "a small car"; "a little (or small) group"*
- S: (adj) little, slight** ((quantifier used with **mass** nouns) small in quantity or degree; not much or almost none or (with 'a') at least some) *"little rain fell in May"; "gave it little thought"; "little time is left"; "we still have little money"; "a little hope remained"; "there's slight chance that it will work"; "there's a slight chance it will work"*

FIGURE 5.6: Results of searching word 'LESS' on Wordnet

- Observe figure 5.6.
- It is showing results of searching word 'LESS' on Wordnet, and word 'MASS' is present in the definition.
- From this we can state that there are good chances of words (like less) having one of words present in glossary (like mass) to be part of its definition but not relevant to it.
- That is why in next approach we've searched for something which can capture the *context of sentence* and it is *word embeddings*.

5.2 Word2Vec

5.2.1 Word embeddings

- A word embedding is a learned representation for text where words that have the same meaning have a similar representation.
- It is this approach to representing words and documents that may be considered one of the key breakthroughs of deep learning on challenging natural language processing problems.
- Word embeddings are in fact a class of techniques where individual words are represented as real-valued vectors in a predefined vector space. Each word is mapped to one vector and the vector values are learned in a way that resembles a neural network, and hence the technique is often lumped into the field of deep learning.
- Key to the approach is the idea of using a dense distributed representation for each word.
- Each word is represented by a real-valued vector, often tens or hundreds of dimensions. This is contrasted to the thousands or millions of dimensions required for sparse word representations, such as a one-hot encoding.

To get the word embeddings, initially we used python libraries Gensim and Spacy.

5.2.2 Gensim

Gensim = “**Generate Similar**” is a popular open source natural language processing (NLP) library used for unsupervised topic modeling[11]. It uses top academic models and modern statistical machine learning to perform various complex tasks such as :

- Building document or word vectors
- Corpora
- Performing topic identification
- Performing document comparison
- Analysing plain-text documents for semantic structure

5.2.3 Spacy

spaCy is a free, open-source library for advanced Natural Language Processing (NLP) in Python[12]. spaCy is designed specifically for production use and helps you build applications that process and “understand” large volumes of text. It can be used to build information extraction or natural language understanding systems, or to pre-process text for deep learning. spaCy provides a one-stop-shop for tasks commonly used in any NLP project, including:

- Tokenisation
- Lemmatisation
- Part-of-speech tagging
- Word-to-vector transformations

5.2.4 Cosine similarity

- **Cosine similarity** measures the **similarity** between two **vectors** of an inner product space. It is measured by the **cosine** of the angle between two **vectors** and determines whether two **vectors** are pointing in roughly the same direction. It is often used to measure document **similarity** in text analysis.
- Now we want to train model which is supposed to consider word (like *moved*) and physics concept interpreted from that word (which is *motion* here) similar and should give high cosine similarity between them.
- Figure 5.7 shows result of cosine-similarities between words by Gensim.
- Figure 5.8 shows result of cosine-similarities between words by Spacy.
- Figure 5.9 shows result of cosine-similarities between clause and words by Spacy.
- Figure 5.10 shows result of Spacy for sample question.


```
similarity between moved and motion is : -0.012911017
similarity between throw and force is : 0.07406683
similarity between look and see is : 0.17175971
```

FIGURE 5.7: Result of cosine-similarities between words given by Gensim

```
Similarity between moved and motion is 0.31523674726486206
Similarity between throw and force is 0.32465919852256775
Similarity between see and look is 0.7343266606330872
```

FIGURE 5.8: Result of cosine-similarities between words given by Spacy

```
Similarity of keywords with clause --> Moved slower in ocean
volume : 0.3407349
work : 0.50075847
motion : 0.4019461
heat : 0.43148878
mass : 0.42658928
```

FIGURE 5.9: Result of cosine-similarities between clause and words given by Spacy

```
Question number : 1
Mike was snowboarding on the snow and hit a piece of ice.
He went much faster on the ice because ____ is smoother.
(A) snow (B) ice
-->
{'wedge', 'equation is solved in spherical coordinates.', 'ice point',
'Weather fronts', '" Linear, surface, volumetric pole density"', 'desert', 'bend'}
```

FIGURE 5.10: Result of Spacy for sample question

5.2.5 Observations

- We've trained the Gensim and Spacy pre-trained model over Wikipedia pages tagged physics and CBSE board science text books from 5th to 10th standard.
- By looking at results from both Gensim and Spacy we can say that both of them are not working the way we wanted.
- This happened because those pre-trained models are trained on general English text and in general English words *throw* and *force* are not similar but *look* and *see* are. Refer figure 5.7 and 5.8.

- Hence now we want model which is **trained over scientific corpora** and which can **capture context of the data** we're giving to it for training.
- Also our glossary have lot of physics concepts which are not useful for us hence we've to make new glossary which is precise collection of physics concepts. Refer figure [5.10](#).

5.2.6 BERT

- BERT (Bidirectional Encoder Representations from Transformers) is a recent paper published by researchers at Google AI Language.
- BERT makes use of Transformer, an attention mechanism that learns contextual relations between words (or sub-words) in a text[13].
- Two separate mechanisms are included in the vanilla form of transformer - an encoder that reads the text input and a decoder that produces a prediction for the task.

5.2.7 The Approach using BERT

Following steps are the approach we've used to get the physics concepts related to given natural language question using BERT :

- *Glossary of physics* : Training questions of QUAREL also contains the quarel relations used to answer them. Quarel relations shows relationship between physics concepts. We have made set of those concepts by parsing quarel relations of each question. A map is made which contains mapping of question to its related physics concepts set (say **Test_Sets**). This has become the testing criteria for each question. Super set of all such sets is our new glossary.
- We've created map (say **Glossary_Map**) which contains mapping of physics concept in glossary to it's word embedding given by BERT.
- Given a natural language question from training data, we are tagging the question using POS-TAG
- Create chunks from question based on RegEx. e.g. <VB> <JJ> <IN><DT> <NN>.
- Each word in chunk is given to BERT which will give it's word embedding.

- Now depending on type of word we're taking weighted average of their embeddings called **avg-vector**(verbs are having weight 100, adjectives are having 50 and others are having 1).
- Then we've taken *cosine similarity* of this **avg-vector** with each vector in **Glossary_Map** and sorted them in descending order.
- Top 4 concepts are chosen for each chunk and added in the set named **Concept_Set**.
- Hence **Concept_Set** is the output set of physics concepts related to question given by BERT.
- Now **Test_Sets** map contains the concepts which we're supposed to find for this question.
- Therefore **Concept_Set** is super-set of the set for question in **Test_Sets** map then we can say that we've achieved our goal.

The following figure 5.11 shows the result of sample question we got using BERT. Although BERT model is capturing the context but it is not trained on

```
The propeller on Kate's boat moved slower in the ocean compared to the river.
This means the propeller heated up less in the (A) ocean (B) river
--->
{'weight', 'heat', 'distance', 'acceleration', 'speed'}
```

FIGURE 5.11: Result of BERT for sample question

scientific text. For these requirements Allen institute of AI have developed BERT model trained on scientific text named Sci-BERT.

5.2.8 Sci-Bert

- SciBERT is a BERT model trained on scientific text.
- SciBERT is trained on papers from the corpus of semanticscholar.org. Corpus size is 1.14M papers, 3.1B tokens. They've use the full text of the papers in training, not just abstracts.
- SciBERT has its own vocabulary (scivocab) that's built to best match the training corpus. They trained cased and uncased versions. They also include models trained on the original BERT vocabulary (basevocab) for comparison[14].

5.2.9 The approach using Sci-BERT

- The only difference between the approach used for BERT model and Sci-BERT is while taking the weighted average of chunks.
- While using BERT we've hard coded the weights (for verb 100, for adjectives 50 and for others 1) but this time we've run the program on several ranges of weights for both verbs and adjectives. The weight for others is kept 1.

The figure 5.12 shows the result of sci-bert for sample question.

```
The propeller on Kate's boat moved slower in the ocean compared to the river.  
This means the propeller heated up less in the (A) ocean (B) river  
Result : {'mass', 'speed', 'brightness', 'friction', 'apparentSize'}
```

FIGURE 5.12: Result of Sci-BERT for sample question

5.3 Results

- For start we've kept verb weights ranging from 100 to 1000 and adjective weights ranging from 50 to 500.
- Adjectives are given lower range because the initial thought was 'By looking at Verbs in sentence we can guess accurately which physics concept is related to the sentence'.
- For example if a sentence contains word *throw* then most probably it is having relation with *force*.
- The figure 5.13 shows heatmap of the f1 scores of results we've got for above range of weights.

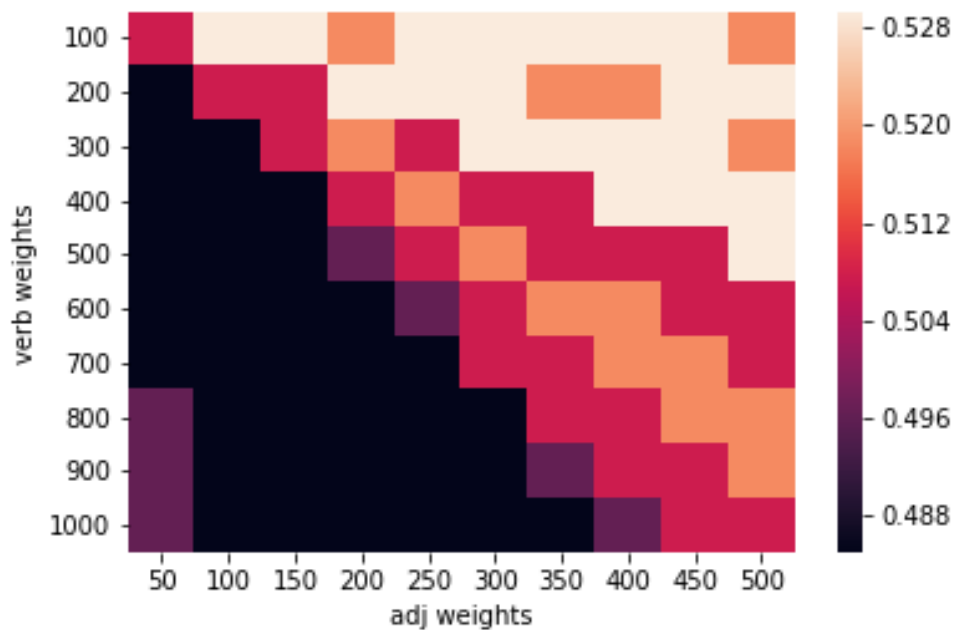


FIGURE 5.13: First heatmap

- Highest f1-score in this is 0.52.
- Observations of the heatmap above says that maximum f1 score is achieved most of the time when verb weight is 100 and adjective weight is in range 100 to 450.
- This shows us that adjectives are also as important as verbs.

- Therefore we've changed the ranges of verb weights and adjective weights for next run. New ranges are verb weights 50 to 5000 and adjective weights 50 to 5000.
- The figure 5.14 shows the heatmap of g1 scores of results we've got for above range of weights.

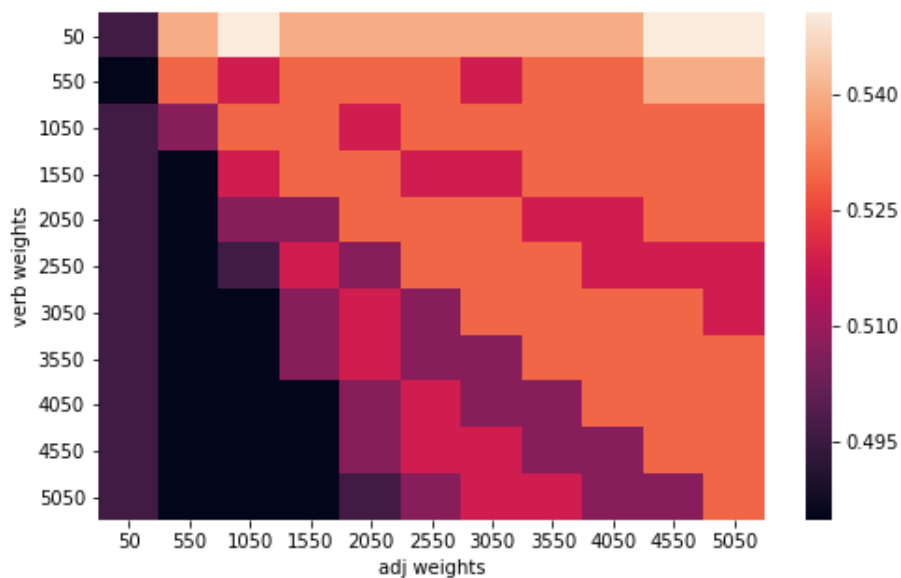


FIGURE 5.14: Second heatmap

- Highest f1-score in this is 0.55.
- Observations of the heatmap above says that maximum f1 score is achieved when verb weight is 50 and adjective weight is above 4000.
- This shows us that instead of keeping high verb weight we should keep high adjective weights.
- Also if we compare the f1 scores for verb weight 50 and verb weight 550. For every value of adjective weight the verb weight 50 is having higher f1 score.
- This shows us that verb weights should be kept low.
- Hence the new ranges for next run are for verb weight 20 to 500 and for adjective weight 4000 to 10000.

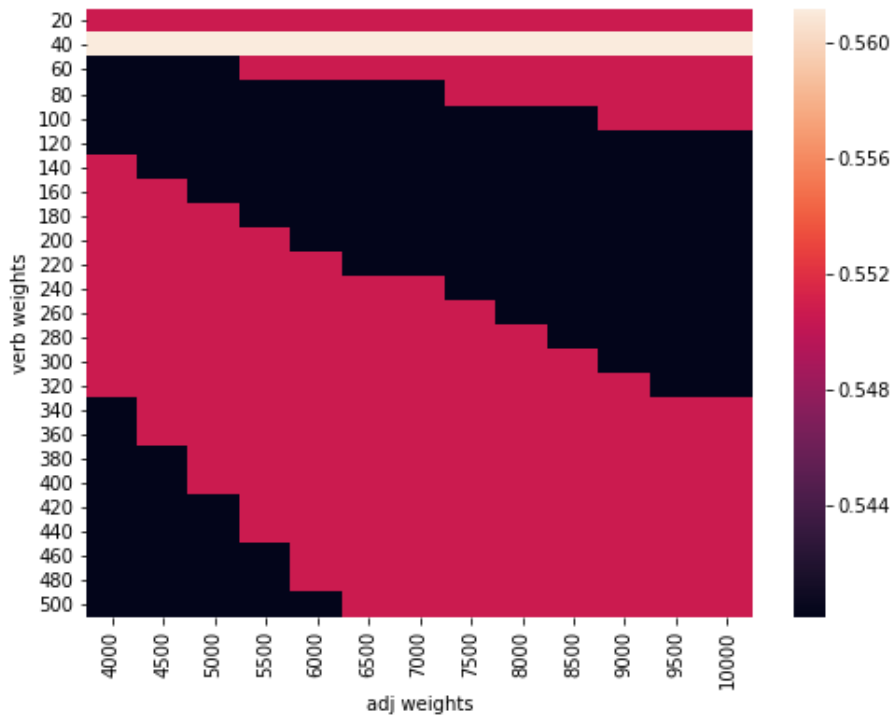


FIGURE 5.15: Third heatmap

- The heatmap above shows that our observations are correct.
- The maximum f1 score is 0.56+
- Maximum f1-score is achieved for verb weight 40 and for every adjective weight.
- The table 5.1 shows results we've got in all approaches explained above.

Approach	Verb weight	Adjective weight	Maximum f1-score
Wordnet	-	-	0.25
Gensim	100	50	0.10
Spacy	100	50	0.10
BERT	100	50	0.40
Sci-BERT	100	50	0.47
Sci-BERT	100-1000	50-500	0.52+
Sci-BERT	50-5050	50-5050	0.55+
Sci-BERT	20-500	4000-10000	0.56+

TABLE 5.1: Results achieved in various approaches

Chapter 6

Conclusion and Future Work

- By observing table 5.1 we can state that adjective weights should be given more weights than verb weights.
- As we can see in figure 5.15 for verb weight 40, every adjective weight is giving us maximum f1 score. Hence that is the area which we may explore further.
- To improve this model's performance, one might change the RegEx we used to extract the chunks from sentence as chunking is the main part which removes unnecessary words from question.
- The only task remaining for completion of stage 2 now is to map question to it's logical form.
- This can be done using quarel relationships that we've acquired in stage 1.
- In stage 3, knowledge graph needs to be built for reasoning purpose. Figure 6.1 shows small portion of the expected Knowledge Graph.

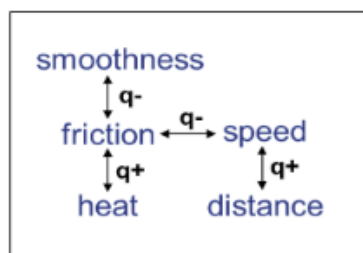


FIGURE 6.1: Small portion of Knowledge Graph

- While building this graph, each physics concept will become node and edges are of two types $q+$ and $q-$ showing direct and inverse relationship between the nodes respectively.
- For reasoning purpose, one need to develop various graph algorithms.

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](https://doi.org/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](https://doi.org/10.18653/v1/d16-1264). URL: <http://dx.doi.org/10.18653/v1/D16-1264>.
- [5] Oyvind Tafjord et al. "QuaRel: A Dataset and Models for Answering Questions about Qualitative Relationships". In: *AAAI*. 2019.
- [6] *Part of Speech (PoS) Tagging*. URL: https://www.tutorialspoint.com/natural_language_processing/natural_language_processing_part_of_speech_tagging.htm. (accessed: 10.07.2020).
- [7] *OpenNLP - Named Entity Recognition*. URL: https://www.tutorialspoint.com/opennlp/opennlp_named_entity_recognition.htm. (accessed: 10.07.2020).
- [8] Leonard Richardson. *Beautiful Soup Documentation*. URL: <https://www.crummy.com/software/BeautifulSoup/bs3/documentation.html>. (accessed: 10.07.2020).
- [9] *Sympy*. URL: <https://www.sympy.org/en/index.html>. (accessed: 10.07.2020).
- [10] Princeton University. *WordNet*. URL: <https://wordnet.princeton.edu/>. (accessed: 10.07.2020).

-
- [11] *Gensim - Introduction*. URL: https://www.tutorialspoint.com/gensim/gensim_quick_guide.htm. (accessed: 10.07.2020).
 - [12] *spaCy 101: Everything you need to know*. URL: <https://spacy.io/usage/spacy-101>. (accessed: 10.07.2020).
 - [13] Rani Horev. *BERT Explained: State of the art language model for NLP*. URL: <https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270>. (accessed: 10.07.2020).
 - [14] *Sci-BERT*. URL: <https://github.com/allenai/scibert>. (accessed: 10.07.2020).