Interpretable Question Answering using Deep Embedded Knowledge Reasoning to Solve Qualitative Word Problems

by

Sanjay Narayana

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved April 2020 by the
Graduate Supervisory Committee:

Chitta Baral, Chair
Arindam Mitra
Saadat Anwar

ARIZONA STATE UNIVERSITY

May 2020

ABSTRACT

One of the measures to determine the intelligence of a system is through Question Answering, as it requires a system to comprehend a question and reason using its knowledge base to accurately answer it. Qualitative word problems are an important subset of such problems, as they require a system to recognize and reason with qualitative knowledge expressed in natural language. Traditional approaches in this domain include multiple modules to parse a given problem and to perform the required reasoning. Recent approaches involve using large pre-trained Language models like the Bidirection Encoder Representations from Transformers for downstream question answering tasks through supervision. These approaches however either suffer from errors between multiple modules, or are not interpretable with respect to the reasoning process employed. The proposed solution in this work aims to overcome these drawbacks through a single end-to-end trainable model that performs both the required parsing and reasoning. The parsing is achieved through an attention mechanism, whereas the reasoning is performed in vector space using soft logic operations. The model also enforces constraints in the form of auxiliary loss terms to increase the interpretability of the underlying reasoning process. The work achieves state of the art accuracy on the QuaRel dataset and matches that of the QuaRTz dataset with additional interpretability.

DEDICATION

I dedicate this to my parents for their constant motivation and encouragement. It goes without saying that this Thesis and my Master's course would not have been possible without their ceaseless support.

ACKNOWLEDGEMENTS

This work would not be complete without mentioning those that helped me throughout. First, I'd like to thank my advisor Dr. Chitta Baral for allowing me to work on my thesis under his mentorship. His valuable guidance and patience allowed me to concentrate on my work without many distractions.

I would be amiss without mentioning the immense role my friend and mentor Dr. Arindam Mitra played in this work. I cannot overstate enough the impact his perseverance, composure, motivation and vast knowledge played in this thesis. His patience is something I greatly acknowledge and appreciate.

I'd also like to thank my peers and seniors - Pratyay Banerjee, Kuntal Pal, Kuldeep Rathore, Swaroop Mishra and Samarth Rawal for their support and for helping me learn various things through discussions, and guiding me when I needed it.

Last but not the least, I wouldn't be able to pursue my dream of doing research without the help and support of my parents, brother and sister-in-law. Their constant support and encouragement helped me stay motivated. I cannot thank my parents enough for all they have sacrificed and done for me.

TABLE OF CONTENTS

APPENDIX

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

Chapter 1

INTRODUCTION AND MOTIVATION

1.1 Introduction

The abundance of user data and computing resources in the past two decades has allowed the field of Artificial Intelligence (AI) to flourish. Popular services like Google Translate and Speech Transcribing services like Siri and Cortana are powered by AI. These techniques are classified under the umbrella term - Natural Language Processing (NLP), a branch of AI that enables machines to process and understand spoken (natural) languages like English and Spanish. The true test of the intelligence of a system is through Question Answering, which involves checking whether it is capable of answering a question by effectively utilizing its underlying knowledge base. Qualitative word problems are a subset of Question Answering that require recognizing and reasoning with qualitative relationships (Tafjord *et al.*, 2018). Relationships in such problems are expressed in a natural language (Eg: If X increases, then so will Y) (Tafjord *et al.*, 2018). To further the research being done in these domains, The Allen Institute of Artificial Intelligence has released two datasets - QuaRTz (Qualitative Relationship Test Set) (Tafjord *et al.*, 2019) and QuaRel (Tafjord *et al.*, 2018).

Recent approaches to solving Question Answering problems involve a multi stage process of converting the input to a logical form, and then later reasoning over this parsed form through a set of logical rules (Mitra *et al.*, 2019). Another approach is to use fine-tune pretrained language models such as BERT (Devlin *et al.*, 2018). Multi-stage models are prone to errors in each stage that accumulate and increase the overall error, whereas approaches such as those that are based on BERT (Devlin *et al.*, 2018) are not interpretable with respect to their reasoning process. This research attempts to embed an interpretable knowledge reasoning approach in a single end-to-end neural model. Therefore, the aim is to improve the accuracy over the current state-of-the-art solutions and to make the approach better interpretable. The approach is applied on Qualitative word problems, and the next two sections briefly describe the two datasets that contain Qualitative word problems.

1.2 Quartz Dataset

The QuaRTz dataset (Qualitative Relationship Test Set) is an open-domain dataset of qualitative relationship questions (Tafjord *et al.*, 2019). An example of an instance from the dataset is shown below. Each problem i contains a 2-way multiple choice question Q_i grounded in a particular situation, and a knowledge sentence K_i that contains general qualitative knowledge to answer the question. Each K_i contains the two concepts being compared and the relationship between them. An example from the dataset is displayed below. In this example, the concepts present in the knowledge passage are *greenhouse gases* - which is highlighted in red - and *ocean acidity* - highlighted in green. The relationship between these two concepts is expressed qualitatively using the words - *increase* and *expand.* Such words / sequence of words that qualitatively convey the intensity (amount) in a qualitative relationship are termed intensifiers. Each Q_i may also contain the concepts and intensifiers with different surface forms. The term *surface form* refers to the description of a concept / intensifier that is expressed in natural language. In the question stem - which is Q_i without the answer options - of the example, the concept *greenhouse gas* is described with a different intensifier *reduced* compared to that present in K_i. The answer options contain the two query intensifiers for the other concept *air quality.* In addition, each

Q_i may also be grounded in one or two different contexts (referred to as worlds here). The example below only contains a single world - *ocean*. Thus the task involves tracking and associating the intensifiers associated with each concept for each of the two worlds. The dataset is split into three partitions : train, dev and test with 2696, 384 and 784 problems in each respectively.

Example:

K: An increase in greenhouse gases will expand the changes that are already being seen including in ocean acidity

Q: what will happen to the acidity of ocean if production of greenhouse gas is reduced?

Options: (A) increase **(B) decrease** [**correct**]

Logical Intepretation:

Q: [**LESS, ”reduced”, ”greenhouse gases”**]

→ (A) [**MORE, ”increase”, ”acidity of the ocean”**]

(B) [**LESS, ”decrease”, ”acidity of the ocean”**]

K: [**MORE, ”increase”, ”greenhouse gases”**]

⇔ [**MORE, ”expand”, ”ocean acidity”**]

1.3 Quarel Dataset

The QuaRel dataset (Tafjord *et al.*, 2018) is similar to the QuaRTz dataset in that it comprises of qualitative relationship problems, but from a fixed set of 19 relationships. An example of an problem from this dataset is shown below. However, unlike the QuaRTz dataset, the QuaRel dataset does not contain annotations for surface forms of the directions and concepts, but instead contains the question interpretation in logical form, and the two different worlds in which the question is grounded. In the example shown below, the left hand side of the logical form represents the current belief in the question. In the example, *qrel(smoothness, lower, world1)* represents that the smoothness of skin - carpet is higher compared to world2 - carpet. The two predicates on the right hand side of the logical form describe the queries corresponding to the two answer options. The logical form of option A denotes that heat is higher, whereas that for option B denotes that heat is lower. The dataset is split into three partitions : train, dev and test with 1941, 278 and 552 problems respectively.

Example:

Q: Tank the kitten learned from trial and error that carpet is rougher then skin. when he scratches his claws over carpet it generates _____ than when he scratches his claws over skin

Options: (A) **more heat** **(B)** less heat

Question Interpretation (Logical Form):

qrel(smoothness, lower, world1)

$\rightarrow$ qrel(heat, higher, world1) ; qrel(heat, lower, world1)

1.4 Research Evaluation

The model will be evaluated on a test set of 784 problems present in the QuaRTz dataset and 552 problems in the QuaRel dataset. As these datasets contain multiple choice problems, the primary evaluation metric will be the accuracy of the predictions on the test set. Since interpretability is also an important feature in our approach, we expect the model to encode different problem-specific attributes based on the expected surface forms.

1.5 Value of the Research

Traditional approaches to solving Question Answering through logical reasoning are mostly non-neural by design. Whereas, recent neural approaches based on the BERT architecture (Devlin *et al.*, 2018) achieve state-of-the-art accuracy but are not interpretable with respect to the reasoning process employed in solving problems. The aim of this research is to explore whether Question Answering can be accomplished and whether its reasoning process can be made transparent in terms of interpretability with the help of constrained learning in a neural setting. Since Qualitative word problems are mainly solved by reasoning over a fixed set of problem-specific attributes, the idea is to construct a neural model that learns the correct values for each property using distributional semantics, and then reasons over these properties to achieve the desired answer.

1.6 Contributions

This work makes the following two contributions:

1. Create a knowledge reasoning framework within a single end-to-end trainable neural network using constrained learning.
2. Make the reasoning process of the model interpretable.

1.7 Structure of the Thesis

This work is structured in the following manner:

1. **Introduction and Motivation** - This chapter contains an overview of the aim

and motivation of this work.

2. **Background and Related Work** - This chapter provides some background on Question Answering and describes existing works that are related to this work.

3. **Methodology** - This chapter outlines the implementation of the various pieces of the approach proposed in this work.

4. **Results** - This chapter presents the various results.

Chapter 2

BACKGROUND AND RELATED WORK

2.1 Question Answering

Question Answering is a branch of Natural Language Processing that involves creating a system capable of understanding and answering a question posed in natural language text. Much of the progress towards Question Answering in recent times is driven by datasets. Different datasets require different methods of reasoning to accurately answer questions. Qualitative problems are a type of Question Answering problems that require reasoning over qualitative relationships. Solving such problems require recognizing the concepts being compared and their intensities (HIGH / LOW, MORE / LESS). To promote research in this domain, The Allen Institute of Artificial Intelligence has released two datasets - QuaRel (Tafjord *et al.*, 2018) and QuaRTz (Qualitative Relationship Test Set) (Tafjord *et al.*, 2019).

Until recently, question answering approaches were categorized to four types (Soares and Parreiras, 2018):

1. **Information Retrieval QA:** Query an unstructured indexed knowledge base by combining the question stem and answer options.

2. **Natural Language Processing QA:** Use machine learning methods to train a model which can recognize certain linguistic properties required to answer a question.

3. **Knowledge Base QA:** Use database style queries to search over structured text.

4. **Hyrbid QA:** Combine the above three methods.

2.2 Related Work

Qualitative problems require models to parse the concepts and their relationships and then reason over these problem-specific attributes. The work creates representations for these attributes through attention mechanisms similar to (Vaswani *et al.*, 2017).

Many semantic based approaches to question answering rely on parsing a given question to a semantic representation, usually termed as a logical form. This logical form is then combined with some form of reasoning either in the form of logical rules, or executed against a database to answer the given question.

The approach followed in (Tafjord *et al.*, 2018) is based on a Neural semantic parser from (Krishnamurthy *et al.*, 2017). This model uses a type-constrained encoder-decoder architecture, where an LSTM is used to encode the question and another LSTM is used to decode the encoded representation into a logical form. Using these parsed logical forms and a set of logical rules, a prolog-style inference engine then determines the correct answer.

Another approach based on semantic parsing for Question Answering is found in (Mitra *et al.*, 2019), which attempts to answer multiple choice questions formed from a passage of text on the life cycle of organisms. This approach uses the semantic parser from (Krishnamurthy *et al.*, 2017) to recognize the question type from a fixed list of question types and then convert the question into a type specific logical form.

Their approach then uses a list of Answer Set Programming rules written in clingo (Gebser *et al.*, 2014) to compute the truth values of each option. The answer option with the higher truth value is scored as the correct answer.

Recently, models based on the BERT architecture (Devlin *et al.*, 2018) have achieved state of the art results on various question answering datasets like SQuAD (Rajpurkar *et al.*, 2016), SWAG (Zellers *et al.*, 2018) and RACE (Lai *et al.*, 2017). Such models are based on the transformer architecture (Vaswani *et al.*, 2017), and are created by pretraining on large corpuses of text. These models can be either be used as features in the word embedding layer, or as base models that can be fine-tuned to a specific task.

2.3 Baseline Models

This section outlines the performance of baseline models on the datasets:

2.3.1 Quartz

The authors of this dataset evaluate 4 baseline models on this dataset. The accuracies of the models are reported in Table 2.1 (Tafjord *et al.*, 2019):

Table 2.1: Accuracies of Baseline Models on the Quartz Test Set

Models ↓	Test Acc.
BERT (IR)	64.4
BERT (Upper Bound)	67.7
BERT-PFT (IR)	73.7
BERT-PFT (Upper Bound)	**79.8**

2.3.2 Quarel

The authors of this dataset evaluate 3 baseline models on this dataset and a custom model called **QUASP+**. The accuracies are present in Table 2.2.

Table 2.2: Accuracies of Various Models on the Quarel Test Set

Models ↓	Test Acc.
IR	48.6
PMI	50.5
BiLSTM	53.1
QUASP+	**68.7**

Chapter 3

APPROACH AND METHODOLOGY

3.1 Problem Formulation

Example 1:

> **K:** An increase in greenhouse gases will expand the changes that are already being seen including in ocean acidity
>
> **Q:** What will happen to the acidity of the ocean if production of greenhouse gas is **reduced** in the ground, compared to the surface rocks, were likely
>
> Options: (A) increase **(B) decrease**

Example 2:

> **K:** The Southern Hemisphere has less trash buildup in the oceans because less of the region is continent.
>
> **Q:** If North America is smaller then Asia, which continent experiences less trash buildup?
>
> **Options: (A) North America (B)** Asia

The task of the QuaRTz dataset and QuaRel dataset is to answer a 2-way question Q_i given a knowledge sentence K_i from a knowledge base K (Tafjord *et al.*, 2019). In the QuaRTz dataset, each K_i contains two concepts - *cause* and *effect* being compared and the qualitative relationship between these concepts expressed in natural language text. However, in the QuaRel dataset, the knowledge passage for a given problem is not explicitly described in natural language. But the concepts being compared and their relationship for a given problem are discernible as explained later in this chapter. In both datasets, each Q_i contains the two concepts of which one concept and its intensifier (amount expressed qualitatively) (high / low) is specified as the setup (given) and the other concept is queried. The concepts mentioned in the question stem are either the same sequence of words as mentioned in the knowledge passage, or they are synonyms expressed in natural language. Each Q_i may be grounded in 0-2 contexts referred to as worlds. The answer options denote either the two possible directions (high / low) of the query concept, or the two possible worlds in which the query concept and its associated intensifier mentioned in Q_i is true. Example 1 defined above serves as an example of the former case, and example 2 for the latter. In the former case, the direction of only the given concept is mentioned, whereas in the latter, the directions of both the given and query concepts are mentioned, and the task is to find in which of the 2 worlds the problem is true. Thus, recognizing a total of 12 attributes is central to solving a qualitative word problem. These attributes are defined below. The first four attributes are present in the knowledge passage, whereas the rest are present in the question stem and answer options. These 12 attributes, and their values for the 2 examples above are listed in Table 3.1.

1. **concept1** - The surface form of the concept that causes an effect or reaction in another concept.

2. **concept2** - The surface form of the concept that is caused a result of concept1.

3. **concept1_direction** - The intensifier and associated surface form of the concept1.

4. **concept2_direction** - The intensifier and associated surface form of the concept2.

5. **given_concept** - The concept whose direction and world are mentioned in the question.

6. **query_concept** - The concept whose direction or world are queried in the question.

7. **given_direction** - The direction word of the given concept.

8. **query_a_direction** - The direction word of the query concept denoted by option (A).

9. **query_b_direction** - The direction word of the query concept denoted by option (B).

10. **given_world** - The world or context in which the the given_concept is specified.

11. **query_a_world** - The world denoted by option (A) for the query_concept.

12. **query_b_world** - The world denoted by option (B) for the query_concept.

Based on the two examples seen above, there are two possible logical forms for problems in the two datasets:

1. (c_1, d_1, c_2, d_2) $(c_{giv}, d_{giv}, w_{giv}) \rightarrow$ (c_a, d_a, w_a) ; (c_b, d_b, w_b) ; d_a != d_b ; $w_a = w_b = w_{giv}$; $c_a = c_b$

2. (c_1, d_1, c_2, d_2) $(c_{giv}, d_{giv}, w_{giv}) \rightarrow$ (c_a, d_a, w_a) ; (c_b, d_b, w_b) ; $d_a = d_b$; w_a != w_b ; $c_a = c_b$

where,

c_1 = concept1,

d_1 = concept1_direction,

c_2 = concept2,

d_2 = concept2_direction,

Table 3.1: Mapping of 12 Attributes for the Two Examples

Attribute ↓	Example 1	Example 2
concept1	greenhouse gases	size of continent
concept2	ocean acidity	trash buildup
concept1_direction	increase (MORE)	less (LESS)
concept2_direction	expand (MORE)	less (LESS)
given_concept	greenhouse gas	size of continent
query_concept	acidity of ocean	trash buildup
given_direction	reduced (LESS)	smaller (LESS)
query_a_direction	increase (MORE)	less (LESS)
query_b_direction	decrease (LESS)	less (LESS)
given_world	ocean	North America
query_a_world	ocean	North America
query_b_world	ocean	Asia

c_{giv} = given_concept,

d_{giv} = given_direction,

w_{giv} = given_world,

$c_a = c_b$ = query_concept,

d_a = query_a_direction,

w_a = query_a_world,

d_b = query_b_direction,

w_b = query_b_world

The first logical form defines the structure of a problem in which the two answer options are two different directions. The *query_a_direction* and *query_b_direction* attributes are different, whereas the *query_a_world* and *query_b_world* attributes are the same. Contrastingly, the second logical form expresses that for problems containing two different worlds as answer options, the *query_a_world* and *query_b_world* attributes are different, whereas *query_a_direction* and *query_b_direction* are the same. The task is to find which among the two queries hold true in the context of the knowledge passage and the given belief from the question for each problem.

3.2 Datasets and Annotations

Each qualitative problem is solved based on the 12 attributes described previously. The model described in section 3.3 is trained to recognize these attributes from a given problem. Both datasets contain annotations for some of these attributes, and these annotations are used to create the required training data. However, the annotations are present in different formats across the two datasets. This section describes the pre-processing or annotation extraction step applied to both datasets to extract the necessary attributes.

3.2.1 Quartz Dataset

As explained earlier, the QuaRTz dataset (Tafjord *et al.*, 2019) contains annotations for both the knowledge passage text and the question stem and options. The annotations are annotated with respect to the two concept types being compared in a problem, which are termed as **cause** and **effect**. The knowledge passage annotations consists of 6 annotated properties -

- **cause_prop** - This property contains the surface form of the cause property in the knowledge passage and corresponds to concept1 from the problem definition.

- **effect_prop** - This property contains the surface form of the effect property in the knowledge passage and corresponds to concept2.
- **cause_dir_str** - This property contains the surface form of the direction string associated with the concept1 in the knowledge passage. It corresponds to concept1_direction in the problem definition.
- **effect_dir_str** - This property contains the surface form of the direction string associated with the concept2 in the knowledge passage and corresponds to concept2_direction.
- **cause_dir_sign** - This property denotes the intensifier (”more” / ”higher” or ”less” / ”lower”) of concept1_direction .
- **effect_dir_sign** - This property denotes the intensifier (”more” / ”higher” or ”less” / ”lower”) of concept2_direction .

The annotations for the question stem and answer options usually contains anywhere between 0 - 8 properties. The dataset suffers from missing annotations for several problems as not all problems may contain these properties. These properties are also annotated with respect to the two concept types - cause and effect.

The annotation fields present in the dataset are:

- **more_cause_prop** - This property denotes the surface form of the cause concept in the direction of ”MORE”.
- **less_cause_prop** - This property denotes the surface form of the cause concept in the direction of ”LESS”

- **more_effect_prop** - This property denotes the surface form of the effect concept in the direction of ”MORE”.
- **less_effect_prop** - This property denotes the surface form of the effect concept in the direction of ”LESS”
- **more_cause_dir** - This property denotes the surface form of the direction of the cause concept in the direction of ”MORE”.
- **less_cause_dir** - This property denotes the surface form of the direction of the cause concept in the direction of ”LESS”
- **more_effect_dir** - This property denotes the surface form of the direction of the effect concept in the direction of ”MORE”.
- **less_effect_dir** - This property denotes the surface form of the the direction of the effect concept in the direction of ”LESS”

3.2.2 Examples of Bad Annotations

Ideally, the reasoning module will reason using the concepts, directions and worlds attributes while answering problems. However, the annotations of the concept attributes or their surface problems from the question stem are unsuitable for use. Below is an example from the QuaRTZ dataset exhibiting such bad annotations of concept attributes from the question stem. The words in red indicate the annotations present in the dataset, whereas the ones in green indicate the desired annotations for this example.

Example:

Q: When someone works out a lot they are _____ than those that never exercise

Options: (A) stronger [**correct**] (B) weaker

Annotations:

less_cause_prop: **those** (**amount of exercise**)

more_effect_prop: **they** (**muscle strength**)

less_effect_prop: **they** (**muscle strength**)

more_cause_prop: **someone** (**amount of exercise**)

As a result, the approach followed in this work is an end-to-end neural model that is capable of doing two tasks:

1. Parse the 8 attributes from a given qualitative problem.

2. Perform reasoning with these attributes to answer the problem.

3.2.3 Annotation Extraction for Quartz Dataset

Solving Qualitative word problems requires the model to recognize the 8 attributes mentioned earlier. The model is trained using supervision to recognize and learn the patterns of these attributes in a given problem. To enable this supervision, the annotations of the QuaRTz dataset as explained previously can be used to extract the required labelled data. The attributes that are extracted from the knowledge passage are the 2 knowledge passage properties mentioned earlier. In addition, the intensifier of the direction strings are also extracted. The task of extracting the attributes from the question is not as straightforward as it is for the knowledge passage. In order to extract the required values for training the model, an algorithm is used to extract the information from each problem.

Algorithm 1: Annotation Extraction of Question in the QuaRTZ dataset.

```
Function extractAnnotationsHelper(is_world_diff, question_part, knowledge_align):
    if is_world_diff then
        check which direction exists from question_part
        set given direction based on answer_key and knowledge_align from options
    else
        set query_a_direction and query_b_direction based on the properties they overlap with
        set given_direction from opposite concept type
    end
Procedure extractAnnotations():
    for each problem do
        question_part = final sentence of question stem
        get is_world_diff
        if options overlap separately with (more_effect_dir, less_effect_dir) or (more_cause_dir, less_cause_dir) but not both then
            extractAnnotationsHelper(is_world_diff)
        else
            set annotations to null
        end
    end
    return
```

An example of the extracted attributes from the algorithm is present in the example below. Since both query A and query B overlap with separately with the *more_* *effect_dir* and *less_effect_dir* annotations, according to the algorithm, the values of these attributes *query_a_direction* and *query_b_direction* are assigned these surface forms. Both these attributes are comparatives of the *effect* concept type. According to the algorithm, the *given_direction* attribute is assigned with **reduced** as the *less_cause_dir* direction annotation belongs to *cause* concept type, which is opposite to the concept type of the query directions. The intensities *LESS* and *MORE* are assigned based on whether there is an overlap with annotations that begin with *more* or *less*.

Example:

K: An increase in greenhouse gases will expand the changes that are already being seen including in ocean acidity

Q: What will happen to the acidity of the ocean if production of greenhouse gas is reduced in the ground, compared to the surface rocks, were likely

Options: (A) increase **(B) decrease**

Annotations:

less_cause_dir: reduced

more_effect_dir: increase

less_effect_dir: decrease

Extracted Attributes:

given_direction: reduced (*LESS*)

query_a_direction: increase (*MORE*)

query_b_direction: decrease (*LESS*)

Algorithm 1 mentions the parameter `is_world_diff`. This variable denotes whether or not two answer options refer to different worlds. A value of 1 denotes that the two answer options refer to different worlds whereas a value of 0 denotes that the both answer options refer to the same world. The *is_world_diff* annotation is used to enforce constraints that are explained later in this chapter. This property is labelled based on whether the two answer options for each problem contain two different worlds. A world is determined as any noun phrase that indicates a named entity. The `question_part` parameter is assigned as the last sentence of the question stem, as most problems in the training data contain the query direction surface form in the final sentence of the question stem. The question stem is split based on the period(.) character to retrieve the final sentence.

3.2.4 Quarel Dataset

Unlike the QuaRTz dataset, the QuaRel dataset does not contain surface forms of the directions and concepts. Also, there is no K_i present in the dataset. However, the logical forms contain the intensifiers and the concepts being compared in a given problem. The logical forms for each question is based on the tuple *(setup, option-A, option-B)* (Tafjord *et al.*, 2018). *setup* here denotes the predicate(s) that describe the current scenario in the question. option-A denotes the predicate of option A, and likewise for option-B. The *setup* predicate is present on the left hand side of the logical form, whereas the option predicates are on the right hand side. There are two possible logical forms in the dataset: (Tafjord *et al.*, 2018)

1. $\text{qrel}(p,\ d,\ w) \rightarrow\ \text{qrel}(p^{'},\ d^{'},\ w^{'})\ ;\ \text{qrel}(p^{''},\ d^{''},\ w^{''})$.

2. $\text{qval}(p,\ d,\ w),\ \text{qval}(p^{'},\ d^{'},\ w^{'''}) \rightarrow\ \text{qrel}(p^{'},\ d^{'},\ w^{'})\ ;\ \text{qrel}(p^{'},\ d^{'},\ w^{'})$.

The predicate *qrel* is used to denote relative values of direction intensities, whereas the *qval* predicate denotes absolute direction intensities.

The knowledge base used for the QuaRel dataset is present in Table 3.2.

The algorithm used to extract the annotations from the QuaRel dataset are explained in Algorithm 2:

Algorithm 2: Algorithm to Extract Annotation from QuaRel Dataset.

```
Procedure extractAnnotations():
    initialize KB of 19 properties
    for each problem do
        answer_option = problem[answerKey]
        logical_form = problem["logical_form"]
        lhs, rhs = logical_form.split(→ )
        concept1, concept2 = lhs["concept"], rhs["concept"]
        knowledge_align = KB[(concept1, concept2)]
        query_a_direction, query_b_direction = rhs[0]["direction"],
         rhs[1]["direction"]
        if answer_option == "A" then
            given_direction = query_a_direction
        else
            given_direction = query_b_direction
        end
    end
    return
```

Table 3.2: The Knowledge Base Used in Constructing Knowledge Sentences for the Quarel Dataset

Property1	Property2	Relationship
Friction	Speed	Inverse
Friction	Distance	Inverse
Friction	Smoothness	Inverse
Friction	Heat	Proportional
Speed	Distance	Proportional
Speed	Smoothness	Proportional
Speed	Heat	Inverse
Distance	Smoothness	Proportional
Distance	Heat	Inverse
Smoothness	heat	Inverse
Distance	Loudness	Inverse
Distance	Brightness	Inverse
Distance	Size	Inverse
Loudness	Brightness	Proportional
Loudness	Size	Proportional
Brightness	Size	Proportional
Speed	Time	Inverse
Time	Distance	Proportional
Weight	Acceleration	Inverse
Strength	Distance	Proportional
Strength	Thickness	Proportional
Mass	Gravity	Proportional
Flexibility	Breakability	Inverse
Exercise	Sweat	Proportional

Example:

K: more weight is caused by less acceleration

Q: the empty cement truck weighed less than the full cement truck so it was _____ when it was accelerating

Options: (A) faster **[correct]** (B) slower

Logical Form:

qrel(weight, lower, empty cement truck) → qrel(acceleration, higher, empty cement truck) ; qrel(acceleration, lower, empty cement truck)

Extracted Attributes:

given_direction: (*LESS*)

query_a_direction: (*MORE*)

query_b_direction: (*LESS*)

In the above example, the concepts are marked in red, the directions are marked in purple and the world is marked in green. The extracted attributes from the algorithm are also mentioned, from the QuaRel dataset, only intensities (*LESS*)) and (*MORE*)) are extractable. As the answer options are two different directions, the *query_a_world* and *query_b_world* attributes have the same value as the *given_world* attribute. This is described in the logical form for the example.

The following sections outline the various parts of the model definition.

3.3 Model Definition

This section outlines the specifics of the model developed in this work.

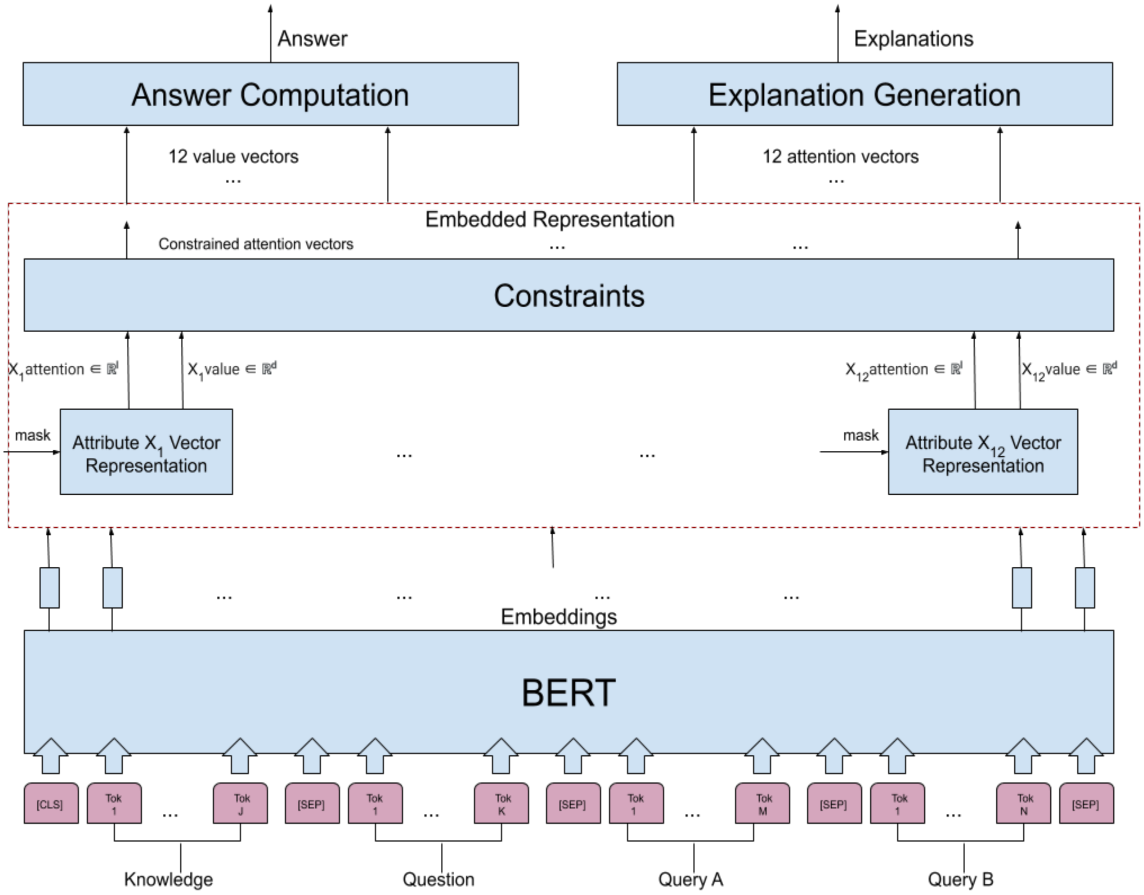
3.3.1 Model Overview

The overview of the model is displayed in Figure 3.1. The various layers of the model are:

1. **Embedding Layer:** Retrieve the embeddings for the input sequence of a qualitative word problem.
2. **Embedded Representation:** Construct the embedded representations for the 12 attributes of a qualitative word problem.
3. **Answer Computation:** Compute the answer for a problem using a differentiable reasoner.
4. **Explanation Generation:** Generate the explanation regarding how the answer was computed by the previous phase.

The process of computing the correct answer includes computing the representations of the problem specific attributes based on distributional semantics and then reasoning using these representations to compute the final answer. To compute the representations of the problem attributes, an embedding layer is first used to obtain the embeddings of the input sequence to the model. The embeddings are obtained using `BERT` (Devlin *et al.*, 2018). The representations for the 12 different attributes are then constructed using these embeddings through 2 vectors -*attention vector* and *value vector*. Constraints are then enforced on the attention vectors of these attributes

Figure 3.1: Model Overview



based on several task specific conditions. A reasoning layer consists of vector based operations on the attribute representations to compute the final answer. The explanation generation layer generates the explanation for the generated answer in terms of the attention vectors of the attributes.

3.3.2 Input Sequence to Retrieve Embeddings

Before performing the required reasoning, the model creates contextual embeddings as features for each problem. The embeddings are created using BERT (Devlin *et al.*, 2018). The code is implemented based on (Wolf *et al.*, 2019). The input sequence fed to the BERT model has the format: *[CLS]* K_i *[SEP]* Q_i *[SEP] Query A [SEP] Query B [SEP]*. Here the CLS and SEP are special tokens denoting the start token and separator tokens of an input sequence specific to the BERT model (Devlin *et al.*, 2018), and it changes depending on the variant of the BERT model that is used. This vector is of length l. Thus, the difference with employing different variations of the BERT architecture in this approach lies only in this layer. To retrieve the embeddings, the implementation also expects the input sequence mask and sequence segment ids. The first part of the sequence *[CLS]* K_i *[SEP]* has segment id zero, whereas the remaining part has segment id of 1.

Input Sequence Sub-Part Masks

Part of the reasoning process involves using different mask vectors to compute the attribute specific attention vectors. The mask vectors passed to the model are listed below. All mask vectors have length l, the same length as the input sequence. These vectors contain a sequence of 1's in specific positions corresponding to the subsequence in which a particular attribute is present.

1. **Knowledge passage mask** : This vector contains a sequence of 1s at positions corresponding to the tokens that constitute K_i in the input sequence. Let $K_i = K_1,\ K_2..\ K_j$ be the sequence of tokens for the Knowledge passage mask of the length j. For the input sequence vector of length l: *[CLS]* $K_1,\ K_2..\ K_j$ *[SEP]* Q_i *[SEP] QueryA [SEP] QueryB [SEP]*. The Knowledge passage mask vector is of the format: $0_{[CLS]}$ 1_{K_1}, 1_{K_2}, .. 1_{K_j} $0_{[SEP]}$ 0_{Q_i} $0_{[SEP]}$ 0_{QueryA} $0_{[SEP]}$ 0_{QueryB} $0_{[SEP]}$

2. **Question stem mask** : This vector contains a sequence of 1s corresponding to the tokens that constitute Q_i in the input sequence.

3. **Option A mask** : This vector contains a sequence of 1s corresponding to the tokens that constitute $QueryA$ in the input sequence.

4. **Option B mask** : This vector contains a sequence of 1s corresponding to the tokens that constitute $QueryB$ in the input sequence.

Direction String Masks

The direction strings of the five different direction attributes that extracted from the annotations are also passed as input to the model in the form of masks. This vector has the same length as the input sequence - l. Similar to the direction value boolean parameters, five additional boolean parameters are used to denote whether direction strings are extracted from the dataset. A value of 1 denotes that the direction strings are present, whereas a value of 0 indicates that the string are not present. Below is a simplified example of a direction string mask vector for the direction word ”reduced” present in the question stem of a problem.

Example:

K: An$_{\mathrm{K1}}$ increase$_{\mathrm{K2}}$ in$_{\mathrm{K3}}$ greenhouse$_{\mathrm{K4}}$ gases$_{\mathrm{K5}}$ will$_{\mathrm{K6}}$ expand$_{\mathrm{K7}}$ the$_{\mathrm{K8}}$ changes$_{\mathrm{K9}}$ that$_{\mathrm{K10}}$ are$_{\mathrm{K11}}$ already$_{\mathrm{K12}}$ being$_{\mathrm{K13}}$ seen$_{\mathrm{K14}}$ including$_{\mathrm{K15}}$ in$_{\mathrm{K16}}$ ocean$_{\mathrm{K17}}$ acidity$_{\mathrm{K18}}$

Q: What$_{\mathrm{Q1}}$ will$_{\mathrm{Q2}}$ happen$_{\mathrm{Q3}}$ to$_{\mathrm{Q4}}$ the$_{\mathrm{Q5}}$ acidity$_{\mathrm{Q6}}$ of$_{\mathrm{Q7}}$ ocean$_{\mathrm{Q8}}$ if$_{\mathrm{Q9}}$ production$_{\mathrm{Q10}}$ of$_{\mathrm{Q11}}$ greenhouse$_{\mathrm{Q12}}$ gas$_{\mathrm{Q13}}$ is$_{\mathrm{Q14}}$ **reduced**$_{\mathrm{Q15}}$

(Query A) increase$_{\text{Query A1}}$ **(Query B)** decrease$_{\text{Query B1}}$

Extracted Attributes:

given_direction: reduced (*LESS*)

query_a_direction: increase (*MORE*)

query_b_direction: decrease (*LESS*)

Input Parameters:

Direction Mask: $[0_{\text{[CLS]}}, 0_{K_1}$ .. $0_{K_{18}}, 0_{\text{[SEP]}}\ 0_{\mathrm{Q1}}\ 0_{\mathrm{Q2}}\ 0_{\mathrm{Q3}}\ 0_{\mathrm{Q4}}\ 0_{\mathrm{Q5}}\ 0_{\mathrm{Q6}}\ 0_{\mathrm{Q7}}\ 0_{\mathrm{Q8}}\ 0_{\mathrm{Q9}}\ 0_{\mathrm{Q10}}\ 0_{\mathrm{Q11}}\ 0_{\mathrm{Q12}}\ 0_{\mathrm{Q13}}\ 0_{\mathrm{Q14}}\ \mathbf{1}_{\mathrm{Q15}}, 0_{\text{[SEP]}}, 0_{\text{Query A1}}, 0_{\text{[SEP]}}, 0_{\text{Query B1}}, 0_{\text{[SEP]}}]$

Direction Intensity: -1 (*LESS*)

World Diff: 0

Direction Intensities

The intensities of the five direction attributes - concept1_direction, concept2_direction, given_direction query_a_direction and query_b_direction are also passed to the model. These intensities are termed $gold\_intensity^{dirX}$. These vectors are of dimension 1. A value of *+1* is used to represent direction *MORE*, and a value of *-1* is used to represent direction *LESS*. If any of these parameters cannot be extracted from the annotations in the dataset, a value of 0 is used. Five additional boolean parameters are used to represent whether or not the five direction value parameters are present for a given problem. A value of 1 for these parameters denotes that the direction values are either 1 or -1, a value of 0 denotes that the direction values are 0. In the example, the surface form of the given_direction attribute - **reduced** is associated with the direction intensity $LESS$, therefore a value of -1 is passed as $gold\_intensity^{given\_direction}$.

World Diff

This parameter is used to indicate whether a problem has different worlds as the different options. A value of 1 indicates that the example contains different worlds as its options or that the attributes query_a_world and query_b_world are different, whereas a value of 0 indicates that the options belong to the same world, or that query_a_world and query_b_world are the same. For the example shown above, the answer options are different directions, hence a value of 0 is assigned to this parameter. These parameters constitute the inputs to the model.

3.3.3 Embedding Layer

To construct the embeddings, the input sequence explained previously - $input_{seq}$ = [CLS] K_i [SEP] Q_i [SEP] $QueryA_i$ [SEP] $QueryB_i$ [SEP] is passed to a BERT-style pretrained model. The pretrained model then outputs the embeddings of the entire input sequence. The dimension of these embeddings is $\mathbb{R}^{l*h}$, where l is the sequence length and h is the hidden size of the pretrained model's hidden states.

$$\begin{gathered}\mathbf{e} = \texttt{BERT}(input_{seq}) \\ \mathbf{e} \in \mathbb{R}^{l*h}\end{gathered} \tag{3.1}$$

3.3.4 Embedded Representation

These embeddings are then fed to 8 different one-layer neural networks independently to compute the attribute representations of the 8 attributes of a qualitative word problem. Each attribute is represented using 2 vectors: *Attention Vector* and *Value Vector*. For each token $i \in 1...l$, where l is the sequence length of the input sequence, each attribute specific network f^X accepts an input vector of dimension h which is the hidden size of a token in the embedding, and outputs a vector of dimension 1 - t_i^X, which denotes the raw attention score or importance of the token in the input sequence with respect to a particular property.

$$t_i^X = f^X(e_i) \tag{3.2}$$

where $f^X : \mathbb{R}^h \rightarrow \mathbb{R}^1$ is a linear function of the form $W^X.e + b^X$, where W^X and b^X are learnable parameters. After obtaining the raw attention scores, the scores are first normalized using a sigmoid operation. To avoid attending to unnecessary tokens, the attention is normalized over different sub-sequence masks for different attributes. The *mask* parameter that is used for normalizing the scores of each attribute is listed in Table 3.3.

$$\begin{gathered}attention^X = sigmoid([t_1^X..t_l^X], mask) \\ attention^X \in \mathbb{R}^l\end{gathered} \tag{3.3}$$

Table 3.3: Normalization Masks for Attributes of a Qualitative Word Problem

Attribute	Value
concept1	K_i
concept2	K_i
concept1_direction	K_i
concept2_direction	K_i
given_concept	Q_i
given_direction	Q_i
given_world	Q_i
query_concept	Q_i
query_a_direction	$Q_i + A_i$
query_a_world	$Q_i + A_i$
query_b_direction	$Q_i + B_i$
query_b_world	$Q_i + B_i$

A weighted sum of the normalized attention scores and the embeddings constitutes the Value vector for each attribute.

$$\begin{gathered} value^X = \sum(attention^X.\mathbf{e}) \\ value^X \in \mathbb{R}^h \end{gathered} \tag{3.4}$$

3.3.5 Constraints

The framework explained so far can also be supplemented with domain and logical constraints specific to Qualitative Word Problems. The constraints that are used here are similar to the constraints used in logical paradigms like Answer Set Programming (Lifschitz, 2008) and First Order Logic (Tannen, 2009). These constraints are implemented similar to those present in (Stewart and Ermon, 2016) and (Xu *et al.*, 2017). The constraints are enforced through several auxiliary loss terms. The loss terms are computed either on the attention vectors or the computed representations of the problem-specific attributes. The various constraints that are enforced in the model are described here.

3.3.6 Direction Spans Constraint

Since the reasoning process relies on recognizing appropriate words as direction words, it makes sense to have the model learn the sequence of words that correspond to the different direction attributes. As a result, an additional constraint imposes a penalty if the attentions of the direction attributes do not match the expected attentions. However, not all problems in the QuaRTz dataset (Tafjord *et al.*, 2019) are completely annotated for directions. This means that care must be taken to only consider those problems that have the required annotations. This constrained is implemented using the *binary_cross_entropy (bce_loss)* function. For each of the 5 direction attributes present in the problem definition, this function is used in the following manner as a loss term:

$$loss\_dir\_span = \mathbf{bce\_loss}(attention^{direction}, mask_{dirX}) \tag{3.5}$$

The five individual loss terms are *loss_concept1_direction_span*, *loss_concept2_direction_span*, *loss_given_direction_span*, *loss_query_a_direction_span*, *loss_query_b_direction_span*. The final loss term *total_loss_dir_span* is the sum of these five terms. An example for the given_direction word **reduced** is present below

$$\begin{aligned} \mathit{total\_loss\_dir\_span} = \mathbf{sum}(&\mathit{loss\_concept1\_direction\_span}, \mathit{loss\_concept2\_direction\_span}, \\ &\mathit{loss\_given\_direction\_span}, \mathit{loss\_query\_a\_direction\_span}, \\ &\mathit{loss\_query\_b\_direction\_span}) \end{aligned} \tag{3.6}$$

Direction Spans Constraint Example:

Q: What$_{\text{Q1}}$ will$_{\text{Q2}}$ happen$_{\text{Q3}}$ to$_{\text{Q4}}$ the$_{\text{Q5}}$ acidity$_{\text{Q6}}$ of$_{\text{Q7}}$ ocean$_{\text{Q8}}$ if$_{\text{Q9}}$ production$_{\text{Q10}}$ of$_{\text{Q11}}$ greenhouse$_{\text{Q12}}$ gas$_{\text{Q13}}$ is$_{\text{Q14}}$ **reduced**$_{\mathbf{Q15}}$

given_direction mask: $[0_{\text{[CLS]}}, .. , 0_{\text{Q1}}, 0_{\text{Q2}}, 0_{\text{Q3}}, 0_{\text{Q4}}, 0_{\text{Q5}}, 0_{\text{Q6}}, 0_{\text{Q7}}, 0_{\text{Q8}}, 0_{\text{Q9}}, 0_{\text{Q10}}, 0_{\text{Q11}}, 0_{\text{Q12}}, 0_{\text{Q13}}, 0_{\text{Q14}}, \mathbf{1_{Q15}}, .., 0_{\text{[SEP]}}]$

attention$^{\textbf{given\_direction}}$: $[0_{\text{[CLS]}}, .. , 0_{\text{Q1}}, 0_{\text{Q2}}, 0_{\text{Q3}}, 0_{\text{Q4}}, 0_{\text{Q5}}, 0_{\text{Q6}}, 0_{\text{Q7}}, 0_{\text{Q8}}, 0_{\text{Q9}}, 0_{\text{Q10}}, 0_{\text{Q11}}, 0_{\text{Q12}}, 0_{\text{Q13}}, 0_{\text{Q14}}, \mathbf{1_{Q15}}, .., 0_{\text{[SEP]}}]$

loss_given_direction_span: 0

3.3.7 Knowledge Passage Directions Disjoint

Since two concepts are always compared with each other in a qualitative word problem, it is straightforward that the corresponding direction words of the concepts are either different words or occupy different positions in the the knowledge passage text if they're the same word. This constraint is enforced with the following equation:

$$\mathit{loss\_disjoint} = \mathit{attention}^{\mathit{concept1\_direction}} * \mathit{attention}^{\mathit{concept2\_direction}} \tag{3.7}$$

The result of this expression is used as a loss term. It returns the measure of similarity between two attention vectors. The higher the similarity, the higher the value returned by the method, and as a result, the higher the value of the loss term that needs to be minimized.

Knowledge Passage Directions Disjoint Example:

K: An$_{\text{K1}}$ **increase**$_{\mathbf{K2}}$ in$_{\text{K3}}$ greenhouse$_{\text{K4}}$ gases$_{\text{K5}}$ will$_{\text{K6}}$ **expand**$_{\mathbf{K7}}$ the$_{\text{K8}}$ changes$_{\text{K9}}$ that$_{\text{K10}}$ are$_{\text{K11}}$ already$_{\text{K12}}$ being$_{\text{K13}}$ seen$_{\text{K14}}$ including$_{\text{K15}}$ in$_{\text{K16}}$ ocean$_{\text{K17}}$ acidity$_{\text{K18}}$

attention$^{\textbf{concept1\_direction}}$: $[0_{\text{[CLS]}}, 0_{\text{K1}}, \mathbf{1_{K2}}, 0_{\text{K3}}, 0_{\text{K4}}, 0_{\text{K5}}, 0_{\text{K6}}, 0_{\text{K7}}, 0_{\text{K8}}, 0_{\text{K9}}, 0_{\text{K10}}, 0_{\text{K11}}, 0_{\text{K12}}, 0_{\text{K13}}, 0_{\text{K14}}, 0_{\text{K15}}, .., 0_{\text{[SEP]}}]$

attention$^{\textbf{concept2\_direction}}$: $[0_{\text{[CLS]}}, 0_{\text{K1}}, 0_{\text{K2}}, 0_{\text{K3}}, 0_{\text{K4}}, 0_{\text{K5}}, 0_{\text{K6}}, \mathbf{1_{K7}}, 0_{\text{K8}}, 0_{\text{K9}}, 0_{\text{K10}}, 0_{\text{K11}}, 0_{\text{K12}}, 0_{\text{K13}}, 0_{\text{K14}}, 0_{\text{K15}}, .., 0_{\text{[SEP]}}]$

loss_disjoint: 0

3.3.8 Constraint on Structure of Qualitative Word Problems

As mentioned earlier in section 3.1, problems from the dataset have two possible logical forms. Recognizing which logical form a problem adheres to is important as the reasoning process depends on whether the given and query directions and worlds are similar. These logical forms are enforced utilizing the is_world_diff annotation. The loss term for this constraint consists of 8 different sub-losses, that together enforce two conditions:

1. Examples where *query_a_world* != *query_b_world* should contain the two different worlds in different answer options.
2. Examples where *query_a_world* == *query_b_world* should contain the two dif-

ferent directions *query_a_direction* and *query_b_direction* in different answer options.

The 8 individual loss terms are explained below. The total loss is the sum of the 8 individual constraints.

1. *query_a_world_in_option*: This term enforces that the $attention^{query\_a\_world}$ vector for examples with *is_world_diff* value of 1 should attend over the query A sub-sequence. It is defined as:

$$query\_a\_world\_in\_option = \sum(1 - (mask_{QueryA}) * attention^{query\_a\_world}) \tag{3.8}$$

2. *query_b_world_in_option*: This term enforces that the $attention^{query\_b\_world}$ vector for examples with *is_world_diff* value of 1 should attend over the query B sub-sequence. It is defined as:

$$query\_b\_world\_in\_option = \sum(1 - (mask_{QueryB}) * attention^{query\_b\_world}) \tag{3.9}$$

3. *query_a_world_not_in_option*: This term enforces that the $attention^{query\_a\_world}$ vector for examples with *is_world_diff* value of 0 should **not** attend over the query A sub-sequence. It is defined as:

$$query\_a\_world\_not\_in\_option = \sum(1 - (mask_{stem}) * attention^{query\_a\_world}) \tag{3.10}$$

4. *query_b_world_not_in_option*: This term enforces that the $attention^{query\_b\_world}$ vector for examples with *is_world_diff* value of 0 should **not** attend over the query B sub-sequence. It is defined as:

$$query\_b\_world\_not\_in\_option = \sum(1 - (mask_{stem}) * attention^{query\_b\_world}) \tag{3.11}$$

5. *query_a_direction_in_option*: This term enforces that the $\mathit{attention}^{query\_a\_direction}$ vector for examples with *is_world_diff* value of 0 should attend over the query A sub-sequence. It is defined as:

$$\mathit{query\_a\_direction\_in\_option} = \sum(1 - (\mathit{mask}_{QueryA}) * \mathit{attention}^{query\_a\_direction}) \tag{3.12}$$

6. *query_b_direction_in_option*: This term enforces that the $\mathit{attention}^{query\_b\_direction}$ vector for examples with *is_world_diff* value of 0 should attend over the query B sub-sequence. It is defined as:

$$\mathit{query\_b\_direction\_in\_option} = \sum(1 - (\mathit{mask}_{QueryB}) * \mathit{attention}^{query\_b\_direction}) \tag{3.13}$$

7. *query_a_direction_not_in_option*: This term enforces that the $\mathit{attention}^{query\_a\_direction}$ vector for examples with *is_world_diff* value of 1 should **not** attend over the query A sub-sequence. It is defined as:

$$\mathit{query\_a\_direction\_not\_in\_option} = \sum(1 - (\mathit{mask}_{stem}) * \mathit{attention}^{query\_a\_direction}) \tag{3.14}$$

8. *query_b_direction_not_in_option*: This term enforces that the $\mathit{attention}^{query\_b\_direction}$ vector for examples with *is_world_diff* value of 1 should **not** attend over the query B sub-sequence. It is defined as:

$$\mathit{query\_b\_direction\_not\_in\_option} = \sum(1 - (\mathit{mask}_{stem}) * \mathit{attention}^{query\_b\_direction}) \tag{3.15}$$

query_a_direction_in_option, World Diff = 0

input$_{\textbf{seq}}$: [CLS] K_1 .. K_{18} [SEP] What$_{\text{Q1}}$ will$_{\text{Q2}}$ happen$_{\text{Q3}}$ to$_{\text{Q4}}$ the$_{\text{Q5}}$ acidity$_{\text{Q6}}$ of$_{\text{Q7}}$ ocean$_{\text{Q8}}$ if$_{\text{Q9}}$ production$_{\text{Q10}}$ of$_{\text{Q11}}$ greenhouse$_{\text{Q12}}$ gas$_{\text{Q13}}$ is$_{\text{Q14}}$ reduced$_{\text{Q15}}$ [SEP]$_{\text{Q16}}$ **increase**$_{\textbf{Query A1}}$ [SEP]$_{\text{Query A2}}$ decrease$_{\text{Query B1}}$ [SEP]$_{\text{Query B2}}$

mask$_{\textbf{QueryA}}$: $[0_{\text{[CLS]}}, \; .. \;, 0_{\text{Q1}}, 0_{\text{Q2}}, 0_{\text{Q3}}, 0_{\text{Q4}}, 0_{\text{Q5}}, 0_{\text{Q6}}, 0_{\text{Q7}}, 0_{\text{Q8}}, 0_{\text{Q9}}, 0_{\text{Q10}}, 0_{\text{Q11}}, 0_{\text{Q12}}, 0_{\text{Q13}}, 0_{\text{Q14}}, 0_{\text{Q15}}, 0_{\text{Q16}}, \mathbf{1}_{\textbf{[Query A1]}}, 0_{\text{[Query A2]}}, 0_{\text{[Query B1]}}, 0_{\text{[Query B2]}}]$

attention$_{\textbf{query\_a\_direction}}$: $[0_{\text{[CLS]}}, \; .. \;, 0_{\text{Q1}}, 0_{\text{Q2}}, 0_{\text{Q3}}, 0_{\text{Q4}}, 0_{\text{Q5}}, 0_{\text{Q6}}, 0_{\text{Q7}}, 0_{\text{Q8}}, 0_{\text{Q9}}, 0_{\text{Q10}}, 0_{\text{Q11}}, 0_{\text{Q12}}, 0_{\text{Q13}}, 0_{\text{Q14}}, 0_{\text{Q15}}, 0_{\text{Q16}}, \mathbf{1}_{\textbf{[Query A1]}}, 0_{\text{[Query A2]}}, 0_{\text{[Query B1]}}, 0_{\text{[Query B2]}}]$

query_a_direction_in_option : 0

query_a_world_not_in_option, World Diff = 0:

input$_{\textbf{seq}}$: [CLS] K_1 .. K_{18} [SEP] What$_{\text{Q1}}$ will$_{\text{Q2}}$ happen$_{\text{Q3}}$ to$_{\text{Q4}}$ the$_{\text{Q5}}$ acidity$_{\text{Q6}}$ of$_{\text{Q7}}$ **ocean**$_{\textbf{Q8}}$ if$_{\text{Q9}}$ production$_{\text{Q10}}$ of$_{\text{Q11}}$ greenhouse$_{\text{Q12}}$ gas$_{\text{Q13}}$ is$_{\text{Q14}}$ reduced$_{\text{Q15}}$ [SEP]$_{\text{Q16}}$ increase$_{\text{Query A1}}$ [SEP]$_{\text{Query A2}}$ decrease$_{\text{Query B1}}$ [SEP]$_{\text{Query B2}}$

mask$_{\textbf{stem}}$: $[0_{\text{[CLS]}}, \; .. \;, \mathbf{1}_{\textbf{Q1}},\mathbf{1}_{\textbf{Q2}}, \mathbf{1}_{\textbf{Q3}}, \mathbf{1}_{\textbf{Q4}}, \mathbf{1}_{\textbf{Q5}}, \mathbf{1}_{\textbf{Q6}}, \mathbf{1}_{\textbf{Q7}}, \mathbf{1}_{\textbf{Q8}}, \mathbf{1}_{\textbf{Q9}}, \mathbf{1}_{\textbf{Q10}}, \mathbf{1}_{\textbf{Q11}}, \mathbf{1}_{\textbf{Q12}}, \mathbf{1}_{\textbf{Q13}}, \mathbf{1}_{\textbf{Q14}}, \mathbf{1}_{\textbf{Q15}}, 0_{\text{Q16}}, 0_{\text{[Query A1]}}, 0_{\text{[Query A2]}}, 0_{\text{[Query B1]}}, 0_{\text{[Query B2]}}]$

attention$_{\textbf{query\_a\_direction}}$: $[0_{\text{[CLS]}}, \; .. \;, 0_{\text{Q1}}, 0_{\text{Q2}}, 0_{\text{Q3}}, 0_{\text{Q4}}, 0_{\text{Q5}}, 0_{\text{Q6}}, 0_{\text{Q7}}, \mathbf{1}_{\textbf{Q8}}, 0_{\text{Q9}}, 0_{\text{Q10}}, 0_{\text{Q11}}, 0_{\text{Q12}}, 0_{\text{Q13}}, 0_{\text{Q14}}, 0_{\text{Q15}}, 0_{\text{Q16}}, 0_{\text{[Query A1]}}, 0_{\text{[Query A2]}}, 0_{\text{[Query B1]}}, 0_{\text{[Query B2]}}]$

query_a_world_not_in_option : 0

Constraint on Query Directions

Another constraint that is helpful in enforcing the logical forms is described here. For examples with *is_world_diff* value of 1, the *query_a_value* and *query_b_value* vectors should ideally represent the same direction word. In order to enforce this constraint through a loss function, the *attention_equal* procedure is used. This function accepts the *attention_query_a_value* and *attention_query_b_value* vectors as input, and returns a higher loss value if the attentions of the two vectors diverge.

The loss term *loss_a_b_equal* is defined below:

$$loss\_a\_b\_equal = \sum(|attention^{query\_a\_direction} - attention^{query\_b\_direction}|) \tag{3.16}$$

3.3.9 Answer Computation - Reasoning

Once the representations of the 8 properties are computed, the next step is to compute the scores for both answer options. The answer with the higher computed score is labeled as the correct answer. The reasoning function calculates the score by comparing the qualitative relationships present in K_i and Q_i. This is done by computing the alignment of the intensities of the two direction attributes in K_i, the alignment between *given_direction* and *query_direction* for the two queries, and also the match between *given_world* and *query_world* for the two queries. These three different indicators are:

- **Knowledge_Direction_Align :** The alignment of intensities between the two attributes $value^{concept1\_direction}$ and $value^{concept2\_direction}$ from the knowledge passage termed *Knowledge_Direction_Align*.

- **Given_Query_Direction_Align :** The alignment of the intensities between

$value^{given\_direction}$ and $value^{query\_a\_direction}$ for option A, termed *Given_Query_A_Direction_Align*, and between $value^{given\_direction}$ and $value^{query\_a\_direction}$ for option B termed *Given_Query_B_Direction_Align*.

- **Given_Query_World_Match :** The alignment between $attention^{given\_world}$ and $attention^{query\_a\_world}$, termed *Given_Query_A_World_Match* ; and between $attention^{given\_world}$ and $attention^{query\_b\_world}$ termed termed *Given_Query_B_World_Match*.

The scores for option A and option B are defined as:

$$\begin{aligned} Query\_A\_Score = {} & Knowledge\_Direction\_Align * Given\_Query\_A\_Direction\_Align \\ & * Given\_Query\_A\_World\_Match \end{aligned} \tag{3.17}$$

$$\begin{aligned} Query\_B\_Score = {} & Knowledge\_Direction\_Align * Given\_Query\_B\_Direction\_Align \\ & * Given\_Query\_B\_World\_Match \end{aligned} \tag{3.18}$$

and,

$$Knowledge\_Direction\_Align = intensity^{concept1\_direction} * intensity^{concept2\_direction} \tag{3.19}$$

$$Given\_Query\_A\_Direction\_Align = intensity^{given\_direction} * intensity^{query\_a\_direction} \tag{3.20}$$

$$Given\_Query\_B\_Direction\_Align = intensity^{given\_direction} * intensity^{query\_b\_direction} \tag{3.21}$$

and,

$$Given\_Query\_A\_World\_Match = 1 \tag{3.22}$$

$$\begin{aligned} Given\_Query\_B\_World\_Match = 1 - \sum(|attention^{query\_a\_world} \\ - attention^{query\_b\_world}|) \end{aligned} \tag{3.23}$$

The variable $indicator^{dirX}$ denotes the intensity of a direction attribute $dirX$. It is predicted by a classifier called **POLARITY_DETECTOR**, which is defined as:

$$\textbf{POLARITY\_DETECTOR} = tanh(W^{POL}.value^{dirX} + b^{POL}) \tag{3.24}$$

POLARITY_DETECTOR : $\mathbb{R}^h \rightarrow \ \mathbb{R}^1 \in [-1, +1]$ is a linear function followed by a tanh activation. Details of this classifier is present section 3.3.9. The value and attention vector of *given_world* and *query_a_world* are assumed to be the same to simplify the **Given_Query_World_Match** indicator variable. As a result the *Given_ Query_A_World_Match* is 1. *Given_Query_B_World_Match* is computed by comparing the attentions of $attention^{query\_a\_world}$ and $attention^{query\_a\_world}$. If the two attentions attend to different tokens, then *Given_Query_B_World_Match* will have a value < 0. This reasoning function is then optimized using cross entropy loss. This constitutes the end-to-end loss - *loss_ce*.

$$loss\_ce = CrossEntropy([Query\_A\_Score, Query\_B\_Score], answer_{index}) \tag{3.25}$$

Direction Align Computation

The two indicator variables - Knowledge_Direction_Align and **Given_Query_Direction_Align** are predicted using the **POLARITY_DETECTOR** classifier. The range of the output of this classifier, termed *predicted_intensity*^{dirX}, is [-1, +1]. values < 0 are treated as the intensity $LESS$, whereas those > 0 are treated as $MORE$. Each of the five direction attributes have an associated intensity with their surface forms that are extracted during the annotation extraction process. This data is used to train this classifier using an L1 loss function applied to the five different direction properties - denoted by $dirX$. The final loss term *loss_dir_align* is the sum of the five individual loss terms. The loss terms for individual directions is defined as:

$$loss\_dir\_align = mean(predicted\_intensity^{dirX} - gold\_intensity^{dirX}) \tag{3.26}$$

The five individual loss terms are *loss_concept1_direction_align*, *loss_concept2_direction_align*, *loss_given_direction_align*, *loss_query_a_direction_align*, *loss_query_b_direction_align*. The final loss term *total_loss_dir_align* is defined as:

$$\begin{aligned} total\_loss\_dir\_align = \mathbf{sum}(&loss\_concept1\_direction\_align, \\ &loss\_concept2\_direction\_align, loss\_given\_direction\_align, \\ &loss\_query\_a\_direction\_align, loss\_query\_b\_direction\_align) \end{aligned} \tag{3.27}$$

The reasoning mechanism defined above is based on the truth table of an equivalent symbolic reasoner displayed in Table 3.4.

Table 3.4: Symbolical Reasoner Truth Table

Knowledge_ Direction_Align	Given_Query_ Direction_Align	Given_Query_ World_Match	Correct Answer?
F	F	F	F
F	F	T	T
F	T	F	T
F	T	T	F
T	F	F	T
T	F	T	F
T	T	F	F
T	F	F	T

3.3.10 Answer Generation - Explainability of Model

As observed so far, a qualitative word problem can be specified by 12 attributes. The problem can be solved by reasoning over these attributes. As the attention vectors of an attribute contain the relevancy scores for each token over the input sequence, the K highest attention scores and the tokens corresponding to the positions of these scores can be combined to visualize the relative contribution of each token in creating the representation of the attribute.

$$Top_K Indices_X = \underset{1..K}{\operatorname{argsort}} \, attention^X$$

$$Top_K Tokens_X = \{token | token \in input_{seq} \ \& \ token_{index} \in Top_K Indices_X\}$$

$$Top_K Score_X = \underset{1..K}{\operatorname{argmax}} \, attention^X$$

In the above set of expressions, the $Top_K Tokens_X$ contains the tokens in the input sequence that have the top-K attention scores for the attribute X. The indices of the tokens with top-K attention scores is denoted by the term $Top_K Score_X$, whereas the scores are denoted by $Top_K Score_X$. In addition, the output of the **POLARITY_DETECTOR** classifier can be used to interpret the intensity of the direction attribute as:

$$Intensity^{dirX} = \begin{cases} > 0 & "MORE" \\ < 0 & "LESS" \end{cases}$$

3.4 Training

For the QuaRTz dataset, the loss function that is to be optimized is the sum of the end-to-end loss and the auxiliary loss terms described till now. However, for the QuaRel dataset, as the gold direction surface forms are not present, two methods are explored:

1. Fine-tune the best model of QuaRTz on the QuaRel dataset. This model is already trained to recognize the direction surface forms from training the QuaRTz dataset.

2. Augment the training data of the QuaRel dataset with the QuaRTz dataset and train it similar to the QuaRTz dataset.

Chapter 4

RESULTS

This chapter outlines the accuracies of the model on the two datasets. As the aim of this work is to also make the model intepretable, the outputs of the attention vectors for the various attributes is also displayed here. The model explained so far is referred to as DEKR (Deep Embedded Knowledge Reasoning).

4.1 Quartz Dataset

Table 4.1 displays the accuracy of the model on 784 problems of the QuaRTz dataset.

Table 4.1: Accuracies of the Model on the Quartz Test Set

Models ↓	Test Acc.
DEKR (Upper Bound)	79.8%

4.1.1 Example of Interpretability

In order to demonstrate the interpretability of the model, this section displays the top 5 tokens with the highest attention for each of the attributes of an example from the test set. The example is described below:

Example:

K: The Southern Hemisphere has less trash buildup in the oceans because less of the region is continent.

Q: If North America is smaller then Asia, which continent experiences less trash buildup?

Options: (A) North America (B) Asia

In the example above, the correct answer is **option (A)**. The attentions of the value and world attributes are displayed below. Since the answer options are two different worlds (North America and Asia), we expect query_a_direction and query_b_direction to attend over the same surface forms, whereas query_a_world and query_b_world should attend over North America and Asia respectively. The comparison of the each the actual and predicated tokens for each of the attributes is described below:

1. **concept1_direction** - The expected surface form for this attribute is **less**, associated with the concept *trash buildup* in the knowledge passage. In the figure below, the token with the highest attention for this attribute is **less**.

2. **concept2_direction** - The expected surface form for this attribute is **less**, associated with the concept *continent* in the knowledge passage. In the figure below, the token with the highest attention for this attribute is **less**.

3. **given_direction** - The expected surface form for this attribute is **smaller**. The token with the highest attention for this attribute is **smaller**.

4. **query_a_direction** - The expected surface form for this attribute is **less**, which is associated with the concept *trash buildup*. In the figure below, the token with the highest attention for this attribute is **less**.

5. **query_b_direction** - The expected surface form for this attribute is **less**, which is associated with the concept *trash buildup*. In the figure below, the token with the highest attention for this attribute is **less**. Since the answer options compare two different worlds, the token with top attention for the claim_a_direction and claim_b_direction is the same.

6. **given_world** - Since the question stem describes that *North America is smaller than Asia*, the expected surface form for this attribute is **North America**. In the figure below, the tokens with the highest attentions for this attribute are **America** and **North**.

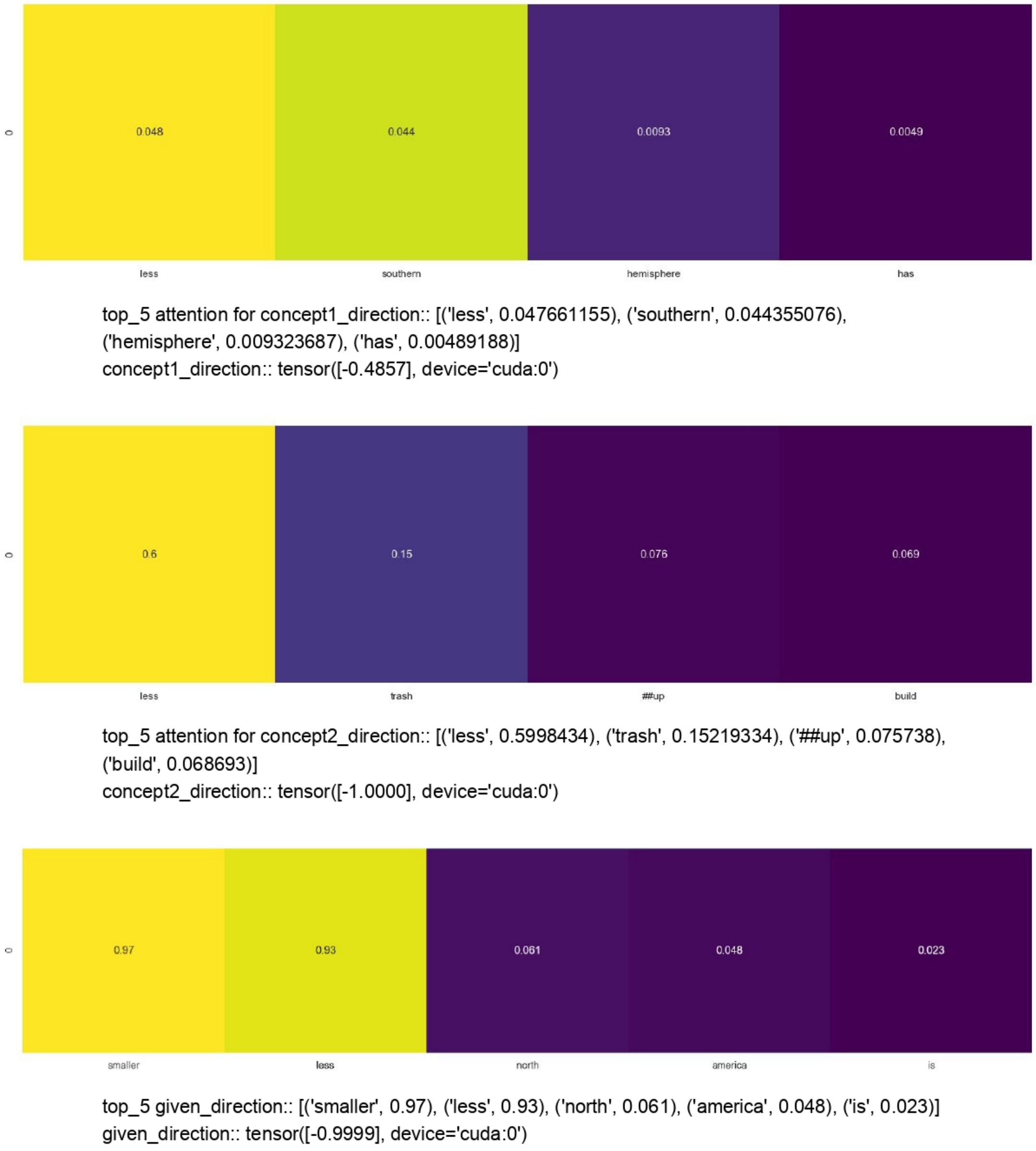
7. **query_a_world** - As answer option (A) denotes a world, the expected surface form for this attribute is **North America**. In the figure below, the tokens with the highest attentions for this attribute are **America** and **North**.

8. **query_b_world** - As answer option (B) also denotes a world, the expected surface form for this attribute is **Asia**. In the figure below, the token with the highest attentions for this attribute is **Asia**.

4.2 Analysis of Flip Questions

One of the characteristics of the QuaRTz dataset is that it contains problems ”flipped” versions of problems. Questions are flipped in a manner so that the correct answer changes. This section demonstrates the capability of the model to answer flipped questions.

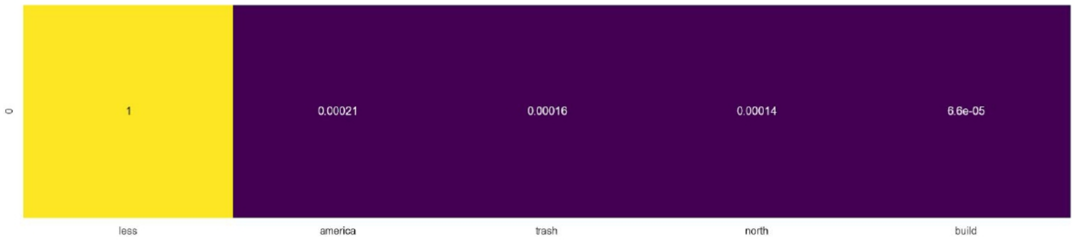
Figure 4.1: QuaRTz Visualization Example



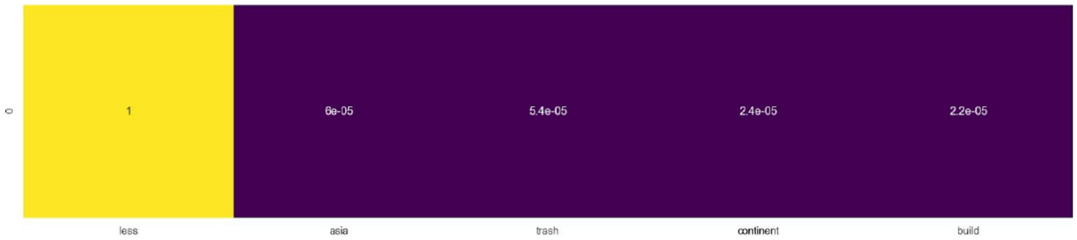
```
top_5 attention for concept1_direction:: [('less', 0.047661155), ('southern', 0.044355076),
('hemisphere', 0.009323687), ('has', 0.00489188)]
concept1_direction:: tensor([-0.4857], device='cuda:0')
```

```
top_5 attention for concept2_direction:: [('less', 0.5998434), ('trash', 0.15219334), ('##up', 0.075738),
('build', 0.068693)]
concept2_direction:: tensor([-1.0000], device='cuda:0')
```

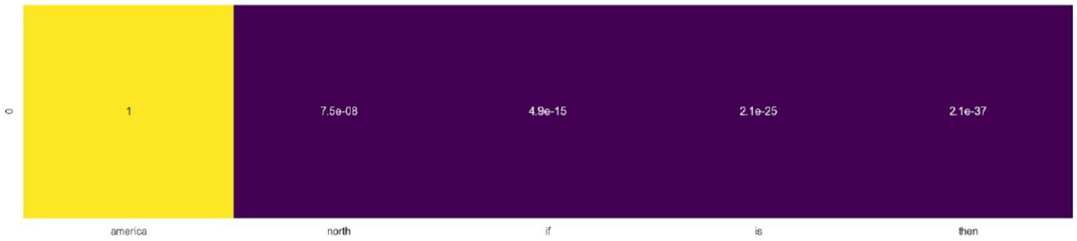
```
top_5 given_direction:: [('smaller', 0.97), ('less', 0.93), ('north', 0.061), ('america', 0.048), ('is', 0.023)]
given_direction:: tensor([-0.9999], device='cuda:0')
```



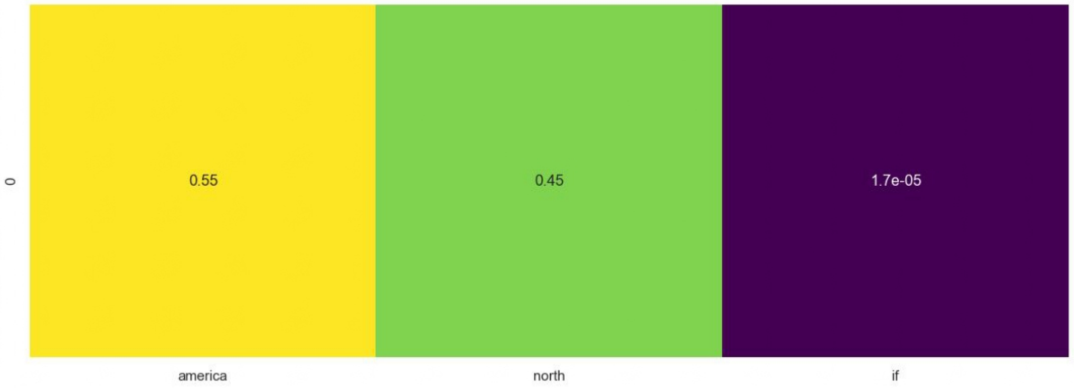
```
top_5 attention for query_a_direction:: [('less', 0.999143), ('america', 0.00020735295), ('trash',
0.00016171749), ('north', 0.00013801016), ('build', 6.647095e-05)]
query_a_direction:: tensor([-0.9999], device='cuda:0')
```



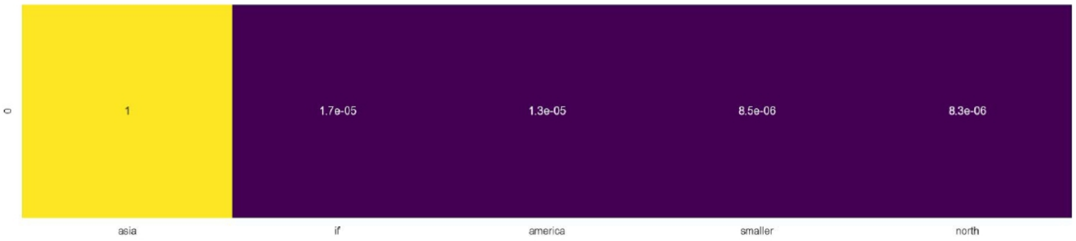
```
top_5 attention for query_b_direction:: [('less', 0.99867404), ('asia', 5.9644906e-05), ('trash',
5.415854e-05), ('continent', 2.352521e-05), ('build', 2.228115e-05)]
query_b_direction:: tensor([-0.9999], device='cuda:0')
```



```
top_5 attention for given_world:: [('america', 0.99999994), ('north', 7.523962e-08), ('if',
4.907268e-15), ('is', 2.1371036e-25), ('then', 2.1084253e-37)]
```



```
top_5 attention for query_a_world:: [('america', 0.55351925), ('north', 0.44641525), ('if',
1.7141238e-05)]
```



```
top_5 attention for query_b_world:: [('asia', 0.999937), ('if', 1.7068656e-05), ('america',
1.2873004e-05), ('smaller', 8.470437e-06), ('north', 8.298614e-06)]
```

Example:

K: An increase in greenhouse gases will expand the changes that are already being seen including in ocean acidity

Q: What will happen to the acidity of the ocean if production of greenhouse gas is reduced in the ground, compared to the surface rocks, were likely

Options: (A) it will increase **(B) it will decrease**

Flipped Example:

K: An increase in greenhouse gases will increase the changes that are already being seen including in ocean acidity

Q: what will happen to the acidity of the ocean if more greenhouse gas gets produced

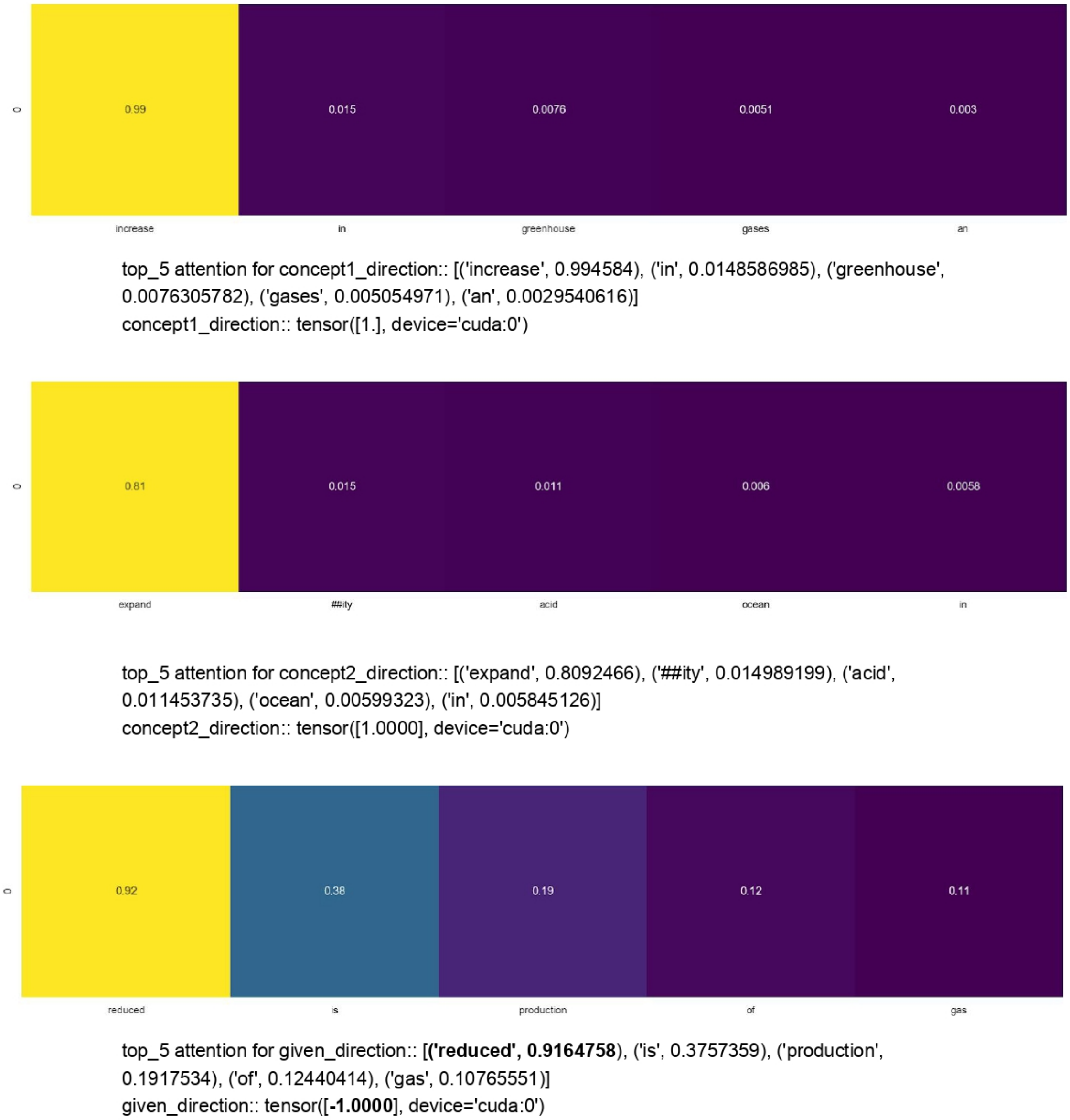
Options: **(A) it will increase** (B) it will decrease

The two examples above two examples from the dev set of the QuaRTz dataset. The two questions have the same knowledge passage, same answer options and similar question stems. In the first example, the greenhouse gases are reduced in the ground, whereas they are more in the flipped example. The model can therefore detect and classify flipped direction words of the dataset. The attentions of the two examples are listed in Figure **??** and Figure **??**.

4.3 Analysis of Negated Comparatives

This section demonstrates the capability of the model to handle negative intensifiers through an example and its flipped version that contains a negative intensifier.

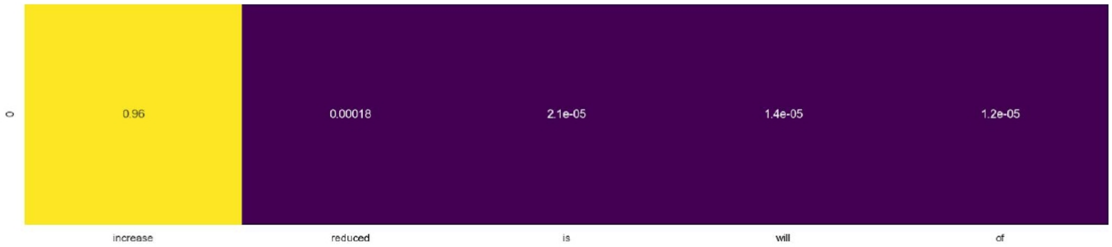
Figure 4.2: Non-Flipped Example



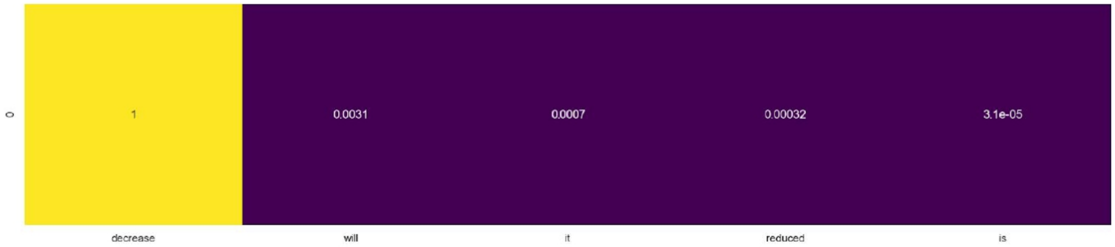
```
top_5 attention for concept1_direction:: [('increase', 0.994584), ('in', 0.0148586985), ('greenhouse',
0.0076305782), ('gases', 0.005054971), ('an', 0.0029540616)]
concept1_direction:: tensor([1.], device='cuda:0')
```

```
top_5 attention for concept2_direction:: [('expand', 0.8092466), ('##ity', 0.014989199), ('acid',
0.011453735), ('ocean', 0.00599323), ('in', 0.005845126)]
concept2_direction:: tensor([1.0000], device='cuda:0')
```

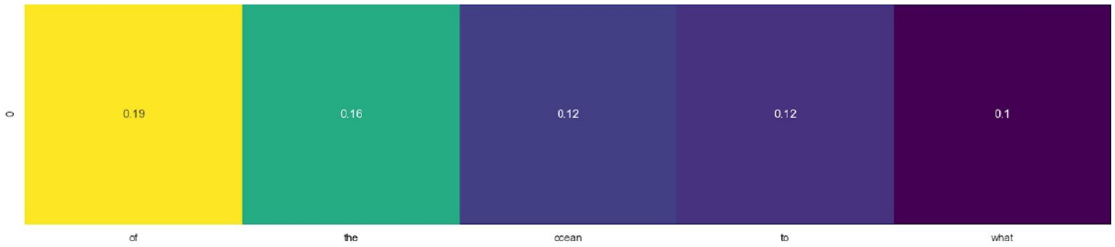
```
top_5 attention for given_direction:: [('reduced', 0.9164758), ('is', 0.3757359), ('production',
0.1917534), ('of', 0.12440414), ('gas', 0.10765551)]
given_direction:: tensor([-1.0000], device='cuda:0')
```



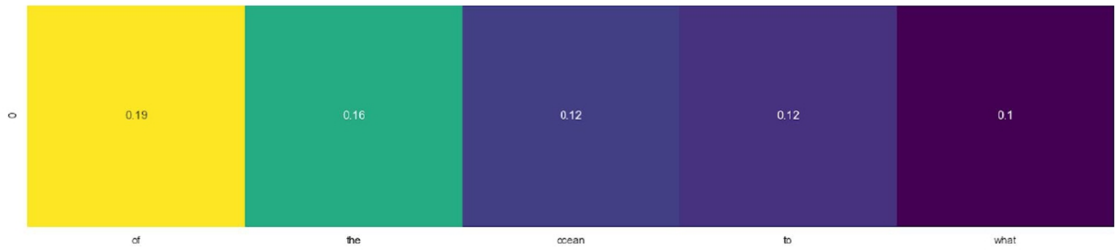
```
top_5 attention for query_a_direction:: [('increase', 0.96477383), ('reduced', 0.00017939364), ('is',
2.116427e-05), ('will', 1.40477705e-05), ('of', 1.2113159e-05)]
query_a_direction:: tensor([1.0000], device='cuda:0')
```



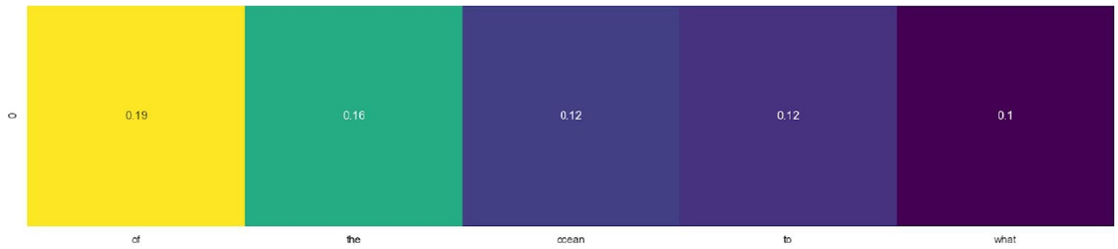
```
top_5 attention for query_b_direction:: [('decrease', 0.99861026), ('will', 0.0030584056), ('it',
0.0007036434), ('reduced', 0.00031624507), ('is', 3.1387422e-05)]
query_b_direction:: tensor([-1.0000], device='cuda:0')
```



```
top_5 attention for given_world:: [('of', 0.19455059), ('the', 0.15957102), ('ocean', 0.12046684), ('to',
0.116634235), ('what', 0.10353842)]
```

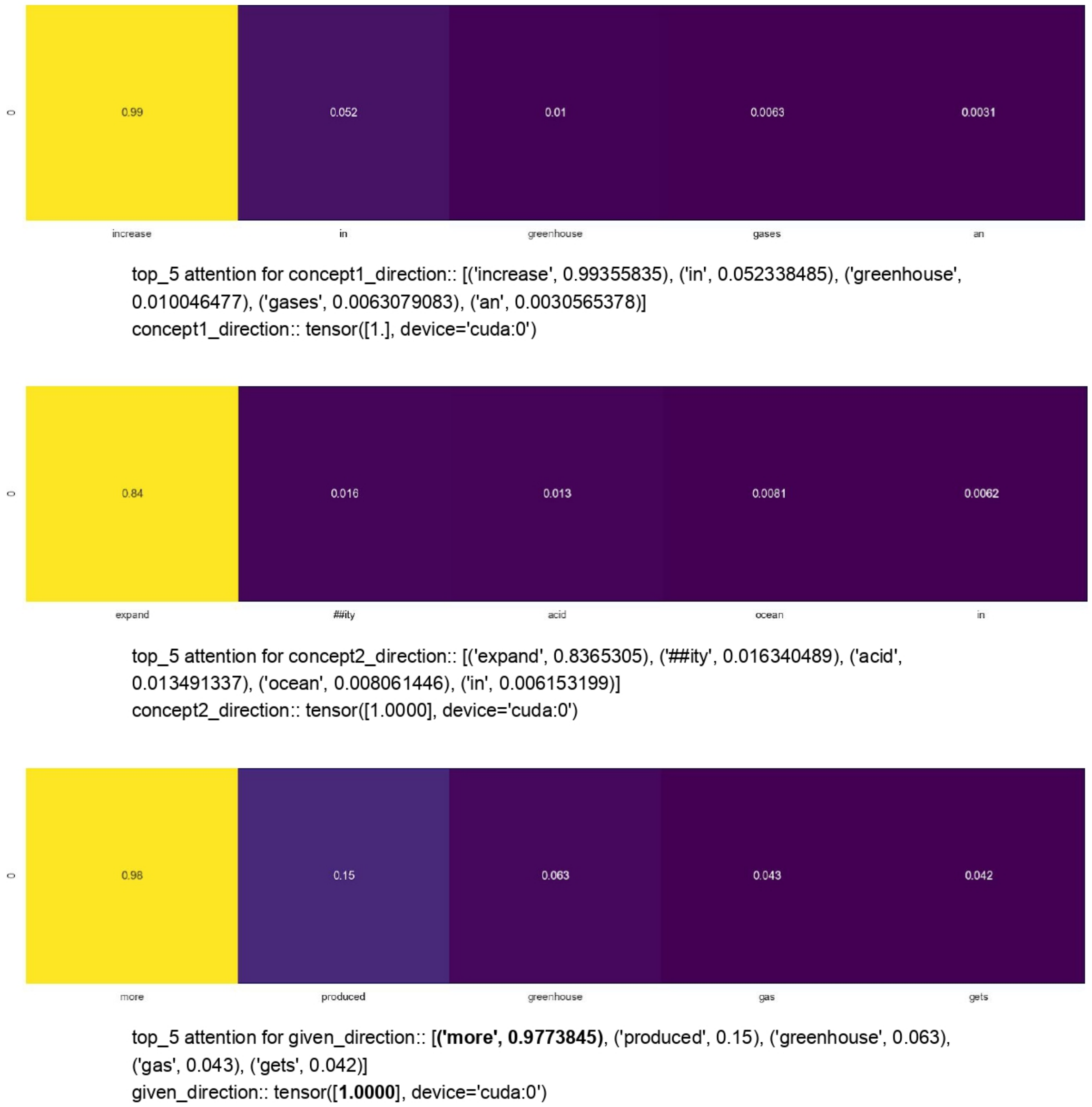


```
top_5 attention for query_a_world:: [('of', 0.19455059), ('the', 0.15957102), ('ocean', 0.12046684),
('to', 0.116634235), ('what', 0.10353842)]
```



```
top_5 attention for query_b_world:: [('of', 0.19455059), ('the', 0.15957102), ('ocean', 0.12046684),
('to', 0.116634235), ('what', 0.10353842)]
```

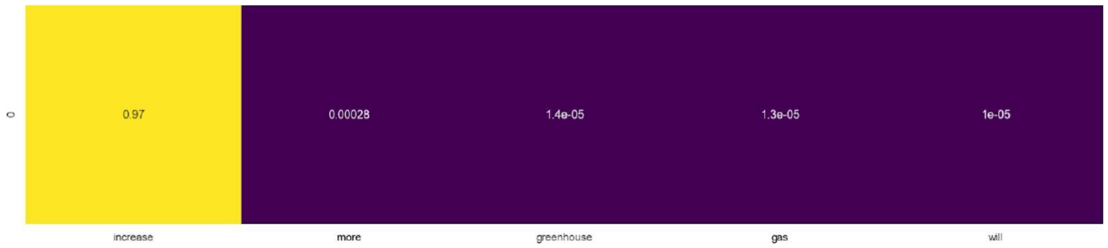
Figure 4.3: Flipped Example



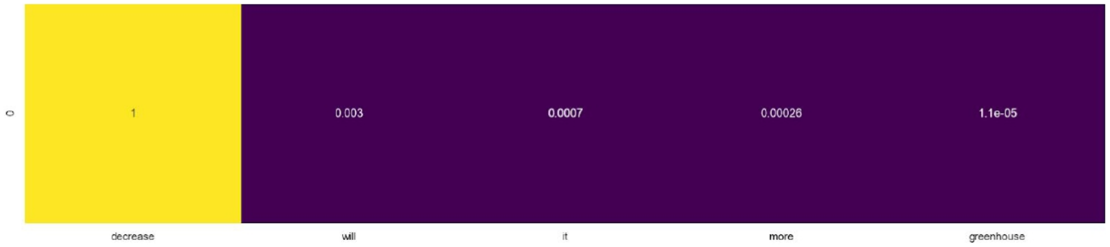
```
top_5 attention for concept1_direction:: [('increase', 0.99355835), ('in', 0.052338485), ('greenhouse',
0.010046477), ('gases', 0.0063079083), ('an', 0.0030565378)]
concept1_direction:: tensor([1.], device='cuda:0')
```

```
top_5 attention for concept2_direction:: [('expand', 0.8365305), ('##ity', 0.016340489), ('acid',
0.013491337), ('ocean', 0.008061446), ('in', 0.006153199)]
concept2_direction:: tensor([1.0000], device='cuda:0')
```

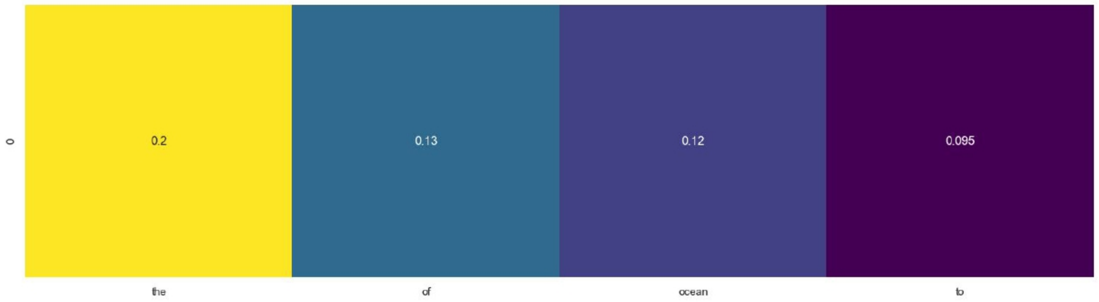
```
top_5 attention for given_direction:: [('more', 0.9773845), ('produced', 0.15), ('greenhouse', 0.063),
('gas', 0.043), ('gets', 0.042)]
given_direction:: tensor([1.0000], device='cuda:0')
```



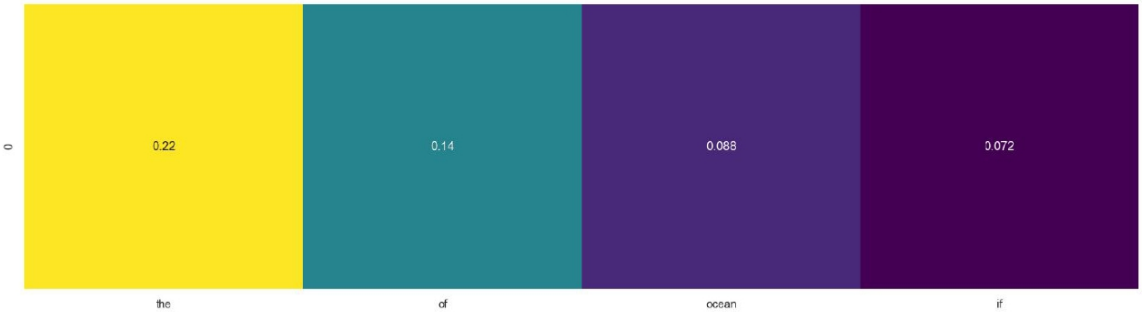
```
top_5 attention for query_a_direction:: [('increase', 0.97464603), ('more', 0.00027501312),
('greenhouse', 1.4331652e-05), ('gas', 1.267379e-05), ('will', 1.0428102e-05)]
query_a_direction:: tensor([1.0000], device='cuda:0')
```



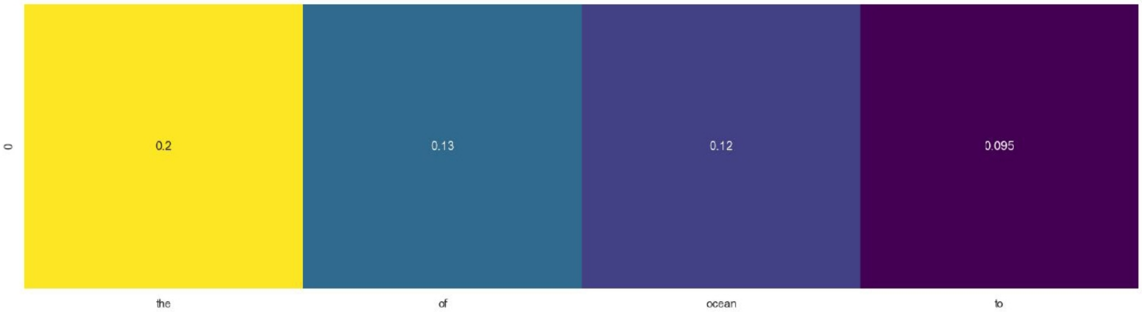
```
top_5 attention for query_b_direction:: [('decrease', 0.9990544), ('will', 0.0030405312), ('it',
0.0006961063), ('more', 0.00026444404), ('greenhouse', 1.066829e-05)]
query_b_direction:: tensor([-1.0000], device='cuda:0')
```



```
top_5 attention for given_world:: [('the', 0.20122848), ('of', 0.131434), ('ocean', 0.116027154), ('to',
0.09549728)]
```

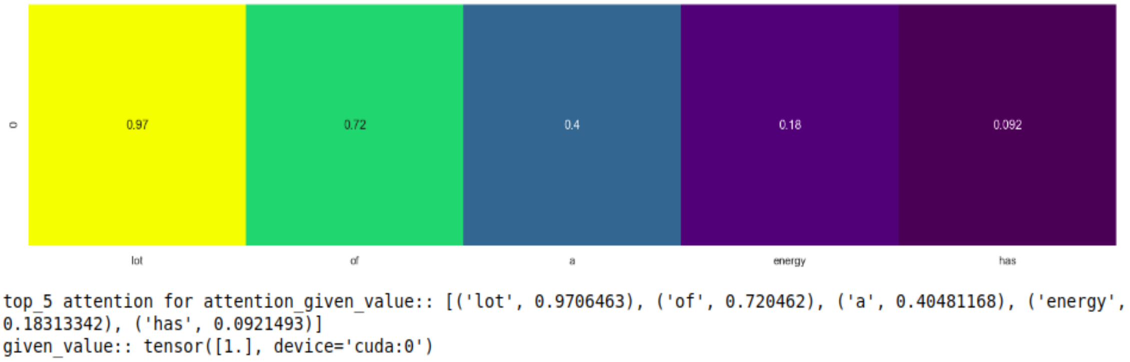


```
top_5 attention for query_a_world:: [('the', 0.20122848), ('of', 0.131434), ('ocean', 0.116027154), ('to',
0.09549728)]
```



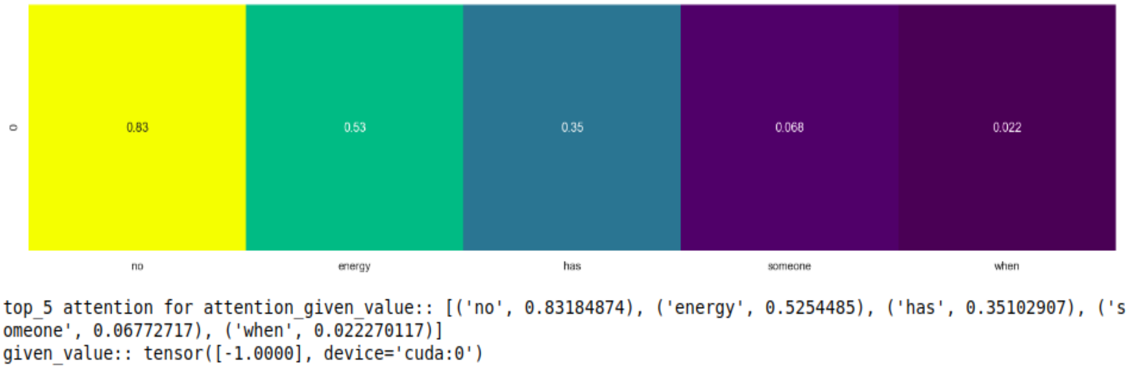
```
top_5 attention for query_b_world:: [('the', 0.21554539), ('of', 0.13646813), ('ocean', 0.08769619),
('if', 0.071778566)]
```

Figure 4.4: Attention of Given_Direction for Example



```
top_5 attention for attention_given_value:: [('lot', 0.9706463), ('of', 0.720462), ('a', 0.40481168), ('energy',
0.18313342), ('has', 0.0921493)]
given_value:: tensor([1.], device='cuda:0')
```

Figure 4.5: Attention of Given_Direction for Example with Negative Intensifier



```
top_5 attention for attention_given_value:: [('no', 0.83184874), ('energy', 0.5254485), ('has', 0.35102907), ('s
omeone', 0.06772717), ('when', 0.022270117)]
given_value:: tensor([-1.0000], device='cuda:0')
```

Example:

K: In erosion, the more energy the water has, the larger the particle it can carry

Q: When someone has a lot of energy they can move something

Options: (A) small **(B) big**

Example of Negated Comparative:

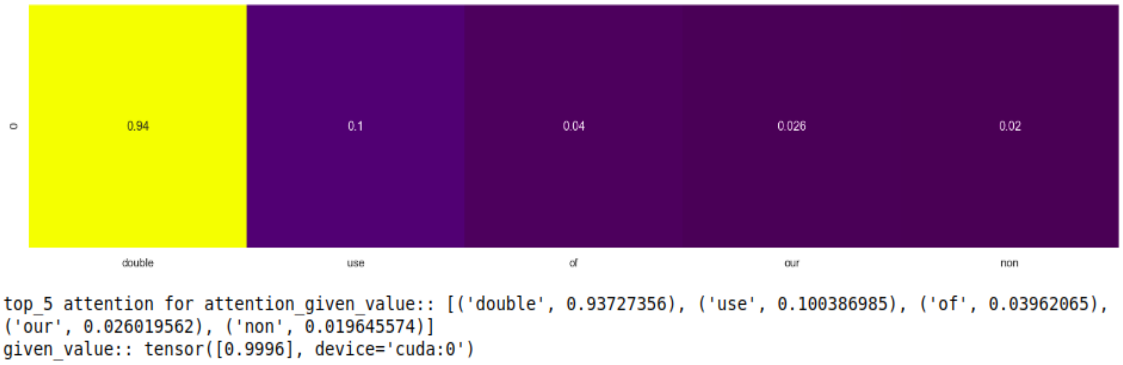
K: In erosion, the more energy the water has, the larger the particle it can carry

Q: When someone has no energy they can move something

Options: **(A) small** (B) big

The intensifier lot of energy from the first example is replaced by the negative intensifier no energy. The model is able to recognize this negated intensifier and appropriately classifies it as ”LESS” as in *less energy*. The attentions of the Given_ Direction attribute for both problems is present in Figure 4.4 and Figure 4.5.

Figure 4.6: Attention of Qualitative Numeric Comparatives



```
top_5 attention for attention_given_value:: [('double', 0.93727356), ('use', 0.100386985), ('of', 0.03962065),
('our', 0.026019562), ('non', 0.019645574)]
given_value:: tensor([0.9996], device='cuda:0')
```

4.4 Handling Qualitative Numeric Comparatives

The model is also able to recognize and detect qualitative numeric comparatives like doubled, halved etc. as direction attributes. An example is present below.

Example:

K: Using less nonrenewable resources means that they will last Longer

Q: If we double our use of nonrenewable resources, we will end up having _____ of them

Options: **(A) less** (B) more

In the above example, the two concepts that are compared are *non renewable resource usage* and *non renewable resource availability.* The *Given_Direction* attribute in Q_i is the word *doubled.* This word has an intensity of "MORE" denoting *more usage of nonrenewable resources.* The model is able to recognize both the surface form and the intensity correctly as seen in Figure 4.6.

4.5 Options with Different Directions but Containing a World

The model is able to recognize the correct surface forms of the *query_a_direction* and *query_b_direction* directions even in the presence of worlds in the answer options. The example below contains a slight modification from the example found in section 1.2. Both the options contain the world *ocean acidity.* But the model is able to recognize the words *increases* and *decreases* as the query direction attributes as shown in Figure 4.7.

Example:

K: An increase in greenhouse gases will increase the changes that are already being seen including in ocean acidity

Q: What will happen to the acidity of the ocean if production of greenhouse gas is reduced in the ground, compared to the surface rocks, were likely

Options: (A) ocean acidity increases **(B) ocean acidity decreases**

4.6 Ablation Analysis on Constraints

This section performs an ablation analysis on the different constraints that are explained in Chapter 3.

where the loss terms are,

ce = Cross Entropy Loss c1 = Knowledge Passage Directions Disjoint

c_dir_span = Direction Spans Constraint

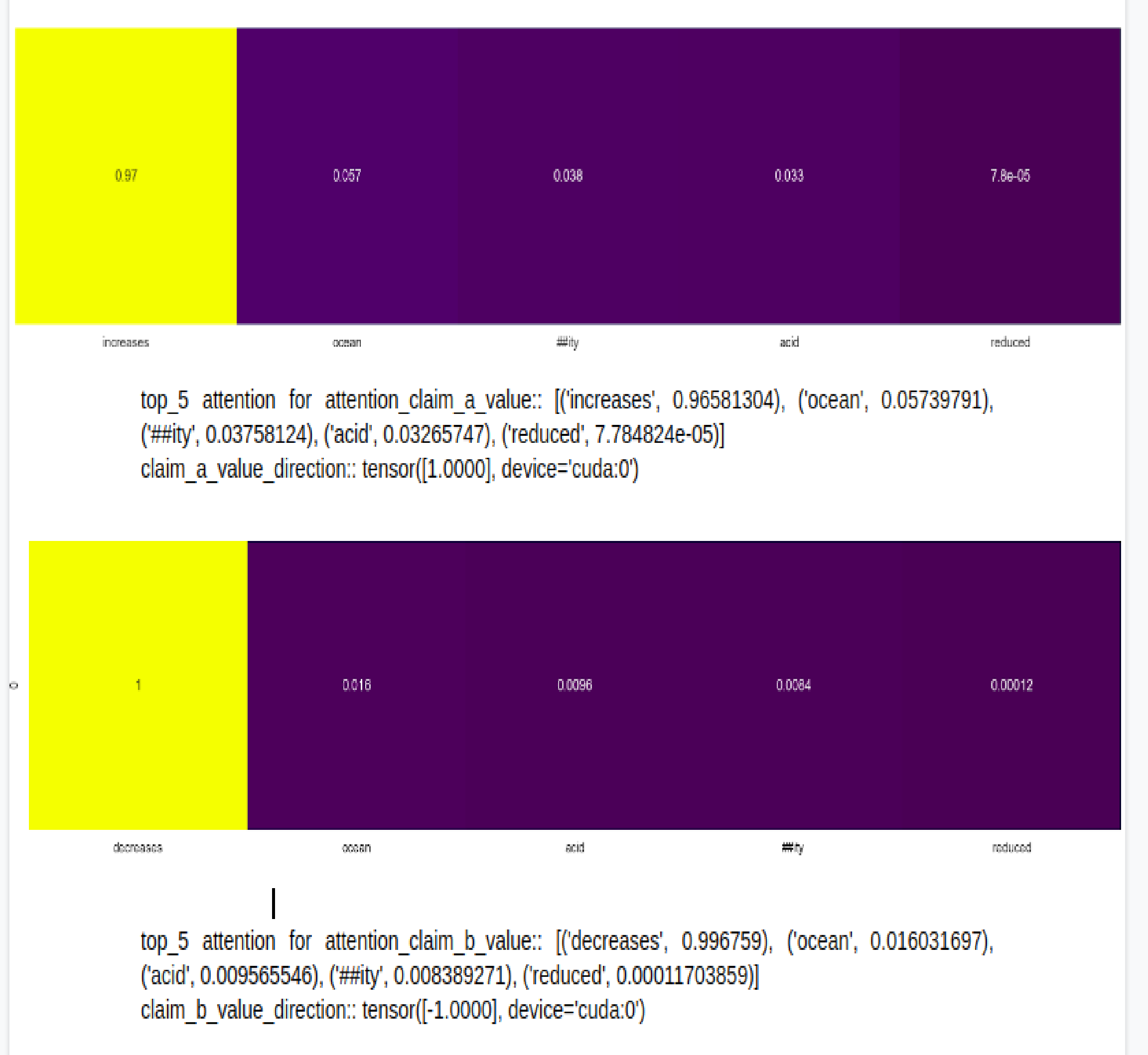
c_dir_align = Direction Align Constraint

loss_struct = Constraint on Structure + Constraint on Query Directions

Table 4.2: Ablation Analysis on Different Loss Terms

	test	concept1_ direction	concept2_ direction	given_ direction	query_a_ direction	claim_b_ direction
ce	50	80	82	50	37	60
ce, loss_ struct	50	78	82	49	37	60
ce, c1	50	19	18	50	62	39
ce, c1, loss_ struct	50	80	82	50	62	39
ce, c1, loss_ struct, c_ dir_align	50	80	88	50	62	39
ce, c1, loss_ struct, c_ dir_align, c_dir_ span	79.48	89	92	80	94	95
ce, dir_ span, dir_ align	78.18	91	88	78	91	94
ce, dir_ span	74.1	16	17	50	50	50

Figure 4.7: Options With Different Directions But Containing a World



```
top_5 attention for attention_claim_a_value:: [('increases', 0.96581304), ('ocean', 0.05739791),
('##ity', 0.03758124), ('acid', 0.03265747), ('reduced', 7.784824e-05)]
claim_a_value_direction:: tensor([1.0000], device='cuda:0')
```

```
top_5 attention for attention_claim_b_value:: [('decreases', 0.996759), ('ocean', 0.016031697),
('acid', 0.009565546), ('##ity', 0.008389271), ('reduced', 0.00011703859)]
claim_b_value_direction:: tensor([-1.0000], device='cuda:0')
```

from the constraint definitions in Chapter 3.

4.7 Quarel Dataset

Table 4.3 displays the accuracy of the model on 552 problems of the QuaRel dataset. The two approaches outlined including using a trained model of the QuaRTz dataset (Tafjord *et al.*, 2019) as the pretrained model and the `bert-large-uncased-whole-word-masking` used as the pretrained model with the training data of the

QuaRel dataset combined with that of the QuaRTz dataset.

Table 4.3: Accuracies of the Model on QuaRel Test Set

Models ↓	Test Acc.
QuaRTZ PFT	81.15%
BERT Augmented	78.98%

4.7.1 Example of Interpretability in Quarel

In order to demonstrate the interpretability of the model with respect to the QuaRel dataset, this section displays the top 5 tokens with the highest attention for each of the attributes of an example from the test set. The example is described below:

Example:

K: less smoothness is caused by more heat

Q: Tank the kitten learned from trial and error that carpet is rougher then skin. when he scratches his claws over carpet it generates _____ then when he scratches his claws over skin

Options: (A) more heat (B) less heat

In the example above, the correct answer is **option (A)**. The attentions of the value and world attributes are present in the figure below. Since the answer options are two different worlds (North America and Asia), we expect claim_a_direction and claim_b_direction to attend over the same surface forms, whereas claim_a_world and claim_b_world should attend over North America and Asia respectively. The comparison of the each the actual and predicated tokens for each of the attributes is described below:

1. **concept1_direction** - The expected surface form for this attribute is **less**, associated with the concept *trash buildup* in the knowledge passage. In the figure below, the token with the highest attention for this attribute is **less**.

2. **concept2_direction** - The expected surface form for this attribute is **less**, associated with the concept *continent* in the knowledge passage. In the figure below, the token with the highest attention for this attribute is **less**.

3. **given_direction** - The expected surface form for this attribute is **smaller**. The token with the highest attention for this attribute is **smaller**.

4. **claim_a_direction** - The expected surface form for this attribute is **less**, which is associated with the concept *trash buildup*. In the figure below, the token with the highest attention for this attribute is **less**.

5. **claim_b_direction** - The expected surface form for this attribute is **less**, which is associated with the concept *trash buildup*. In the figure below, the token with the highest attention for this attribute is **less**. Since the answer options compare two different worlds, the token with top attention for the claim_a_direction and claim_b_direction is the same.

6. **given_world** - Since the question stem describes that *North America is smaller than Asia*, the expected surface form for this attribute is **North America**. In the figure below, the tokens with the highest attentions for this attribute are **America** and **North**.

7. **claim_a_world** - As answer option (A) denotes a world, the expected surface form for this attribute is **North America**. In the figure below, the tokens with the highest attentions for this attribute are **America** and **North**.

8. **claim_b_world** - As answer option (B) also denotes a world, the expected surface form for this attribute is **Asia**. In the figure below, the token with the highest attentions for this attribute is **Asia**.

4.8 Error Analysis

This section outlines categories of errors for the misclassified examples from the dev set for the QuaRTz dataset (Tafjord *et al.*, 2019). Out of 384 examples in the dev set, 87 examples were misclassified. The counts of the various categories are present in Table 4.4

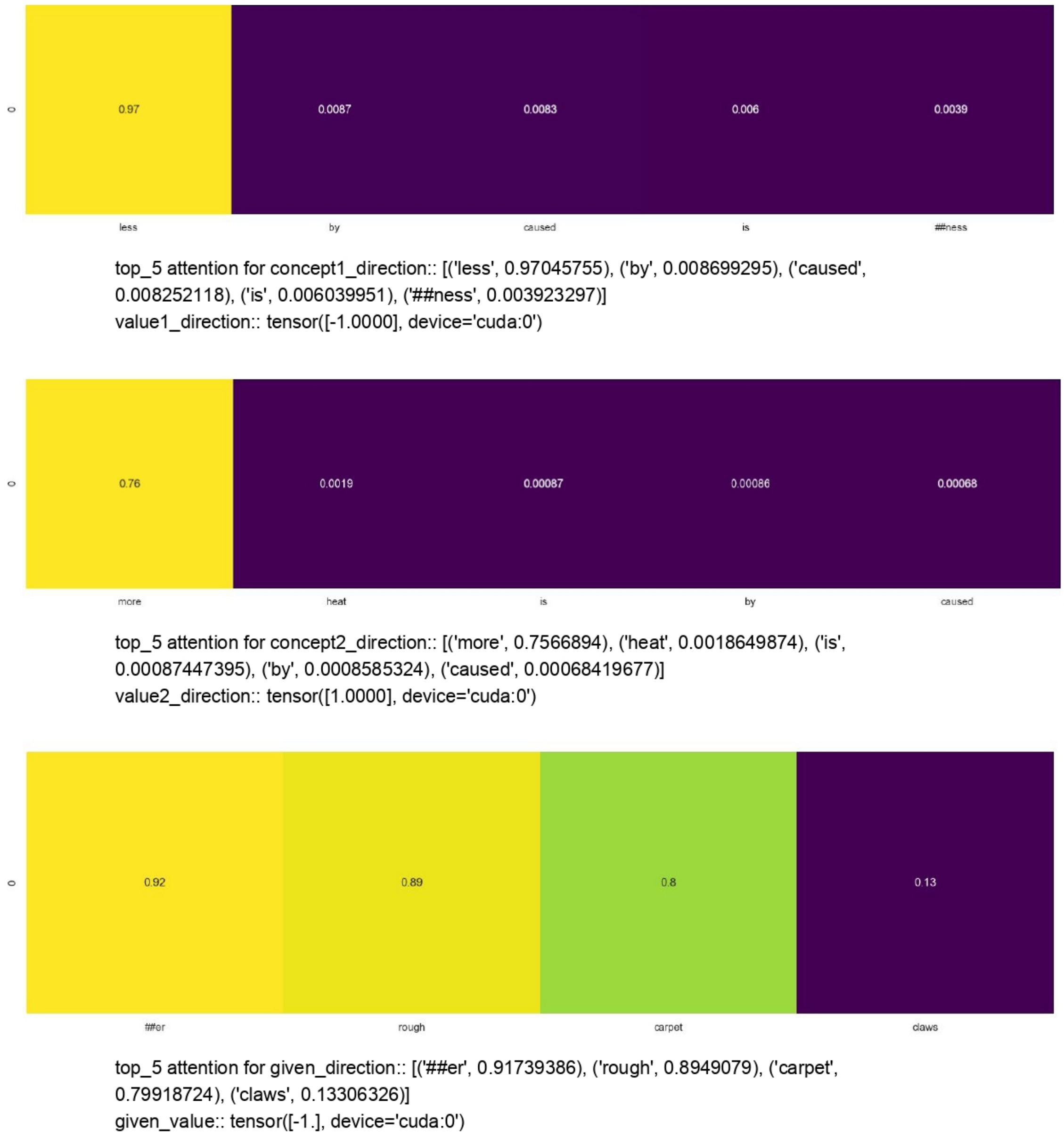
Table 4.4: Counts of Failure Examples from Quartz Dataset

Categroies ↓	Count
Incorrect Value Prediction	35
Wrong Word Detected as the Value	28
Numbers	9
Given World Is Equal to Claim World	6
Dataset Errors	5
Commonsense Reasoning	4

4.8.1 Incorrect Value Prediction

Some direction words like **younger**, **closer**, **farther** etc., can refer to either *MORE* or *LESS* direction intensities based on the concept they are associated with in a particular problem. This ambiguity in terms of direction intensity is the major reason for errors in the QuaRTz dataset (Tafjord *et al.*, 2019). Below is an example

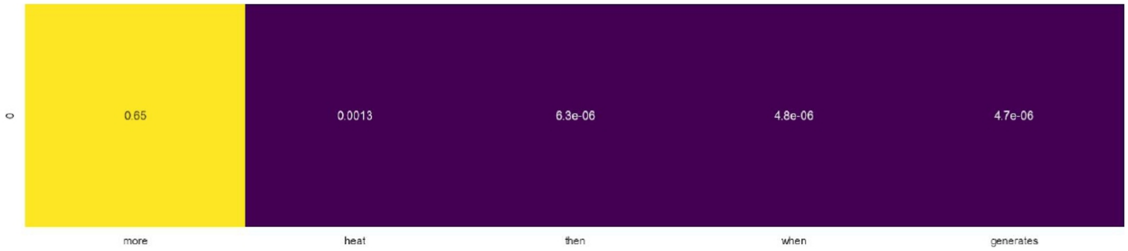
Figure 4.8: QuaRel Example Visualization



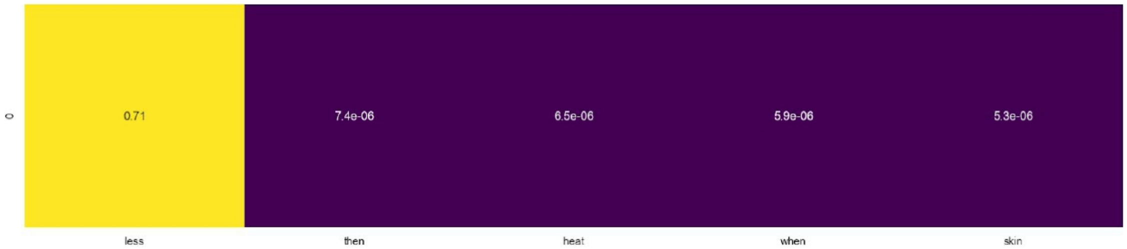
```
top_5 attention for concept1_direction:: [('less', 0.97045755), ('by', 0.008699295), ('caused',
0.008252118), ('is', 0.006039951), ('##ness', 0.003923297)]
value1_direction:: tensor([-1.0000], device='cuda:0')
```

```
top_5 attention for concept2_direction:: [('more', 0.7566894), ('heat', 0.0018649874), ('is',
0.00087447395), ('by', 0.0008585324), ('caused', 0.00068419677)]
value2_direction:: tensor([1.0000], device='cuda:0')
```

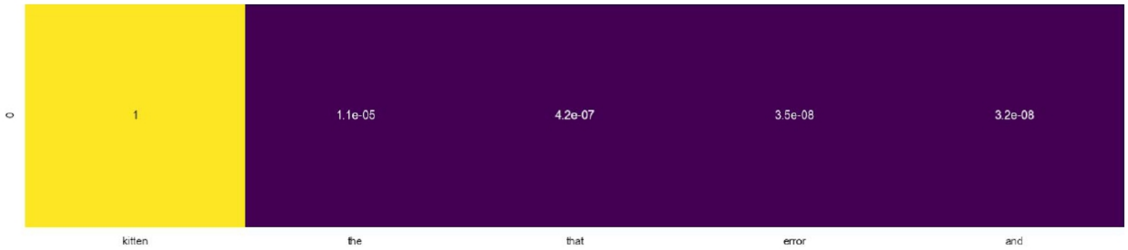
```
top_5 attention for given_direction:: [('##er', 0.91739386), ('rough', 0.8949079), ('carpet',
0.79918724), ('claws', 0.13306326)]
given_value:: tensor([-1.], device='cuda:0')
```



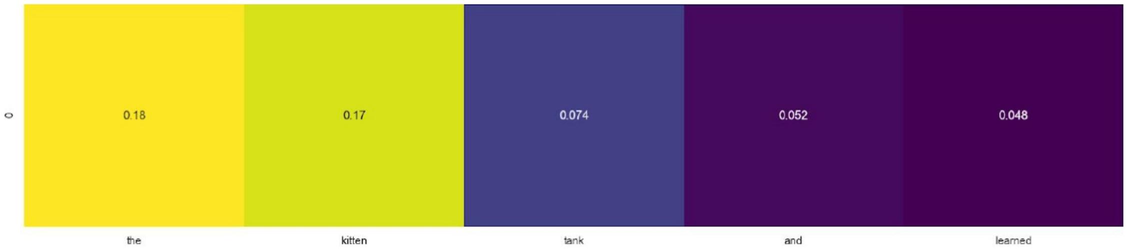
```
top_5 attention for query_a_direction:: [('more', 0.6467518), ('heat', 0.0012980526), ('then',
6.346475e-06), ('when', 4.7536646e-06), ('generates', 4.7350377e-06)]
query_a_direction:: tensor([1.0000], device='cuda:0')
```



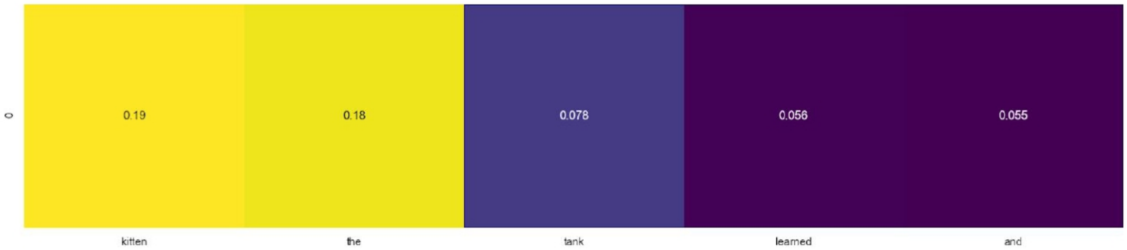
```
top_5 attention for query_b_direction:: [('less', 0.70727277), ('then', 7.41439e-06), ('heat',
6.548007e-06), ('when', 5.914423e-06), ('skin', 5.2569e-06)]
query_b_direction:: tensor([-1.0000], device='cuda:0')
```



```
top_5 attention for given_world:: [('kitten', 0.99998856), ('the', 1.0967166e-05), ('that',
4.2317185e-07), ('error', 3.541252e-08), ('and', 3.248346e-08)]
```



```
top_5 attention for query_a_world:: [('the', 0.18330696), ('kitten', 0.17491995), ('tank', 0.0744242),
('and', 0.051707607), ('learned', 0.048058078)]
```



```
top_5 attention for query_b_world:: [('kitten', 0.18627007), ('the', 0.18287793), ('tank', 0.077515945),
('learned', 0.055615216), ('and', 0.054710947)]
```

of a problem from the dev set for which the model incorrectly predicted the sign of the direction word **farther**.

K: When particles of matter are closer together, they can more quickly pass the energy of vibrations to nearby particles

Q: If jim moves some particles of matter **farther** apart, what will happen to the rate at which they can pass vibrations on to nearby particles

Options: (A) decrease (B) increase

Here, the two concepts being compared are **energy of vibrations** and **how close particles are**. Since the second concept is *how close particles are* and not the *distance between particles*, the word farther in the problem has direction intensity *LESS* as in particles are ”less closer to each other”. As the word **farther** is more often associated with the direction ”more” in the training data, option B is incorrectly misclassified as the answer.

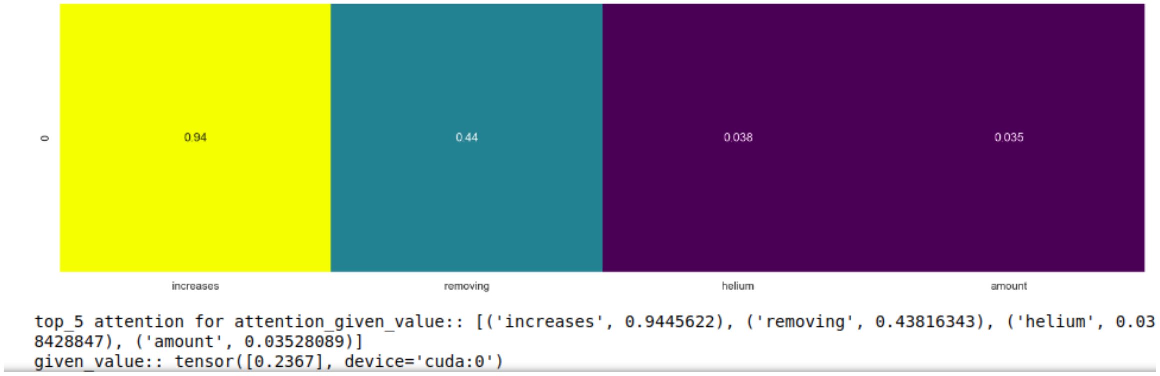
4.8.2 Wrong Word Detected as the Value

For some examples, the incorrect word can be detected as one of the direction attributes, leading to incorrect prediction. This is demonstrated in the below example:

K: When particles of matter are closer together, they can more quickly pass the energy of vibrations to nearby particles

Q: If mona is removing helium from a balloon and she increases the amount she is removing, what happens to the amount of energy the helium particles can pass amongst each other

Options: (A) decreases (B) increases



```
top_5 attention for attention_given_value:: [('increases', 0.9445622), ('removing', 0.43816343), ('helium', 0.03
8428847), ('amount', 0.03528089)]
given_value:: tensor([0.2367], device='cuda:0')
```

Figure 4.9: Example of a Problem With Incorrect Given Value

In the above example, the concepts being compared are **energy of vibrations** and **how close particles are**. The alignment between these concept is proportional. The word **removing** in the question stem should ideally be detected as the *given_value* attribute and should be used to infer that the amount of helium in the balloon is less. However, 4.8.2 shows the top 5 attention tokens of the *given_value* attribute. The token **increases** has higher attention compared to the required word **removing**. As a result, the *given_value_attribute* is associated with the direction ”more”, and option (B) is incorrectly predicted as the answer.

4.8.3 Numbers

The model can also incorrectly answer problems which contains direction attributes in the form of numbers. With numbers as direction words, it is difficult to identify its absolute magnitude and thereby its direction sign. An example is presented below.

Example:

K: Many diseases become more common as people grow older.

Q: If mona is currently 35 years old and she lives her life until the age of 65, what happens to her chance of contracting a disease as she grows older

Options: (A) increase **(B) decrease**

In the above problem, the *given_value* attribute is 65 as Mona lives until the age of 65. It is classified as ”more” by the model, whereas it ought to have been classified as ”less” in order to correctly answer this problem. Such problems highlight the difficulty of handling numbers as directions.

4.8.4 Given World Is Equal to Claim World

In order to compute the match score between the *given_world* and *claim_world* attributes, the *given_world* attribute is set to be the same as the *claim_a_world* attribute. As a result, the model incorrectly predicts the answer for some problems with the following logical structure:

$(c_{giv},\, d_{giv},\, w_{giv}) \rightarrow\ (c_a,\, d_a,\, w_a)\ ;\, (c_b,\, d_b,\, w_b)\ ;\, d_a = d_b\ ;\, w_a\ != w_b\ ;\, w_{giv} = w_b.$

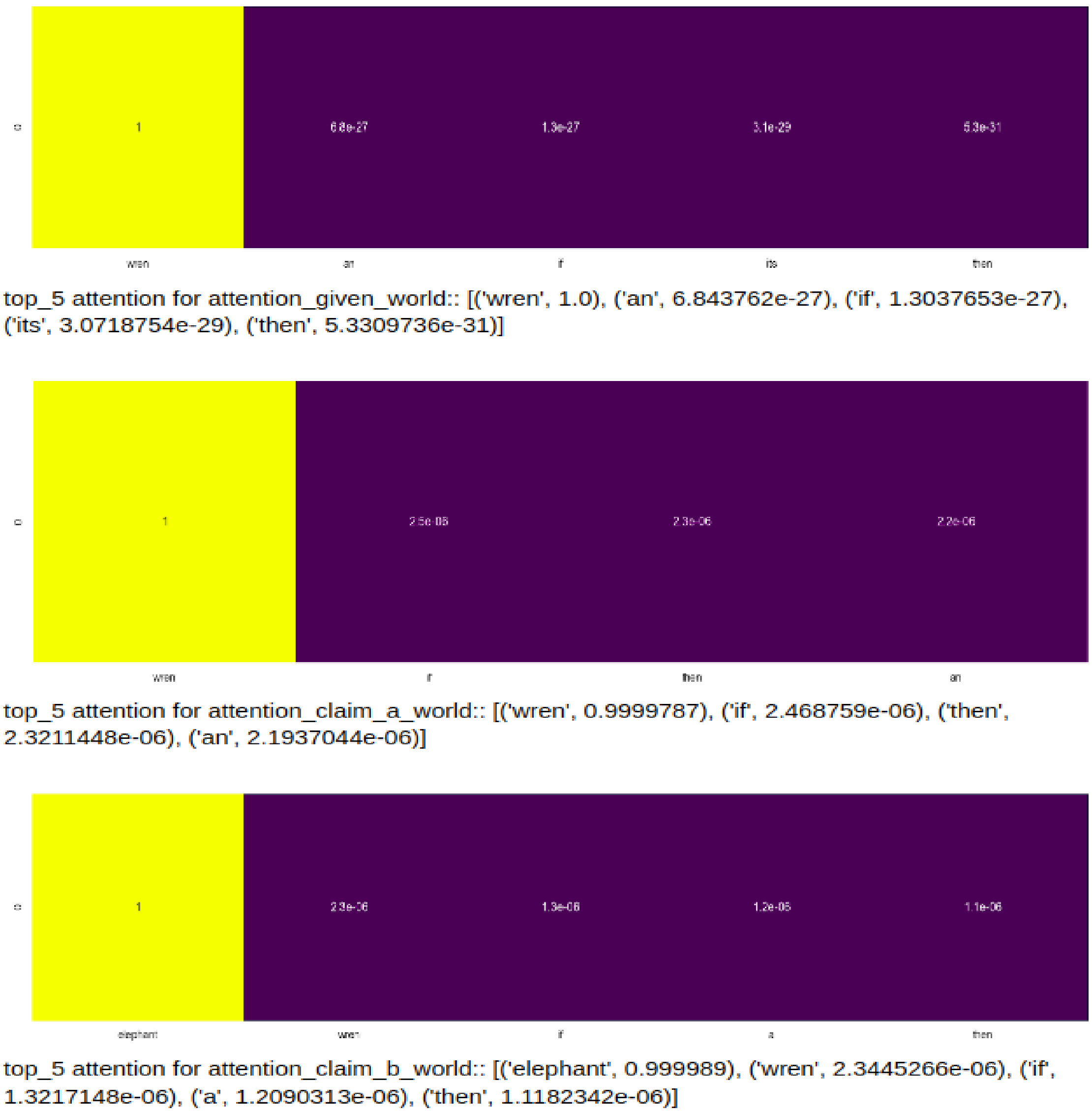
If the question stem describes the *given_direction* and *given_world* attributes from the perspective of *claim_b_world*, then the world match is incorrectly computed, as one of the assumptions of the model is to set *given_world* as *claim_a_world*. This is demonstrated in the example below:

Example:

K: As blood volume in the body increases, blood pressure increases.

Q: If an elephant has more blood in its body then a wren, which animal has higher blood pressure?

Options: (A) Wren **(B) Elephant**



```
top_5 attention for attention_given_world:: [('wren', 1.0), ('an', 6.843762e-27), ('if', 1.3037653e-27),
('its', 3.0718754e-29), ('then', 5.3309736e-31)]
```

```
top_5 attention for attention_claim_a_world:: [('wren', 0.9999787), ('if', 2.468759e-06), ('then',
2.3211448e-06), ('an', 2.1937044e-06)]
```

```
top_5 attention for attention_claim_b_world:: [('elephant', 0.999989), ('wren', 2.3445266e-06), ('if',
1.3217148e-06), ('a', 1.2090313e-06), ('then', 1.1182342e-06)]
```

Figure 4.10: Example of a Problem With Incorrect Given World

In the above problem, the *given_direction* attribute attends to the token **more** corresponding to the concept - *blood volume.* The *claim_a_world* and *claim_b_world* attributes attend to **Wren** and **Elephant** respectively, and *given_world* attends to **Wren** as its representation is computed as the same as *claim_a_world.* However, since *Elephant* has more blood and not *Wren* in the given problem, the world match scores are incorrectly computed, and option (A) is predicted as the answer incorrectly. The attentions of the three world attributes are shown in 4.8.4.

4.8.5 Dataset Errors

The last category of errors of the model are the inescapable dataset errors. One example of a problem that was inaccurately predicted because of an error in the label of the correct answer is presented below.

Example:

K: The greater the mass of an object, the more matter it contains

Q: Steve knows that the mass of joe's car is greater than the mass of flo's car. he concludes that _____ contains more matter than the other

Options: (A) Joe's car **(B) Flo's car**

In the above example, the concepts being compared are the **the mass of an object** and **the amount of matter**. The relationship between these concepts is proportional. In the question stem, Joe's car is known to possess more mass than Flo's car. Therefore, Joe's car should contain more matter than Flo's car, and the correct answer should be option (A). But option (B) was marked as the correct answer by the labelers of the dataset.

4.8.6 Commonsense Reasoning

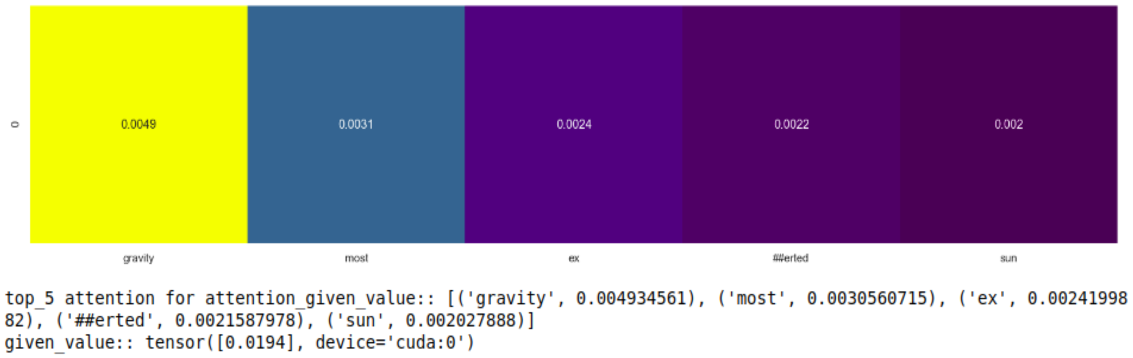
Another category of problems the model sometimes fails is when the problem requires commonsense reasoning to accurately predict the answer. An example of such kind of problems is presented below.

Example:

K: Objects that are closer together have a stronger force of gravity.

Q: Which planet has the most gravity exerted on it from the sun?

Options: **(A) Mercury (B)** Mars



```
top_5 attention for attention_given_value:: [('gravity', 0.004934561), ('most', 0.0030560715), ('ex', 0.00241998
82), ('##erted', 0.0021587978), ('sun', 0.002027888)]
given_value:: tensor([0.0194], device='cuda:0')
```

Figure 4.11: Example of a Problem That Requires Commonsense Reasoning

In the above problem, the model requires commonsense that the distance between the Sun and Mercury is less than the distance between the Sun and Mars. As there is no explicit mention of which object is closer to the Sun in the question stem, the token *gravity* is incorrectly detected as the *given_value* attribute. 4.8.6 shows the top 5 attentions tokens for this attribute.

4.9 Analysis of Earlier Approaches

Details of an alternate approach that did not yield satisfactory accuracy is detailed in Appendix A. The reasoning engine in this alternate approach was based on approximations of the AND, OR and NOT logical gates in vector space. However, this approach did not perform better than random accuracy, but the results improved once the truth value computation was made differentiable through the use of the **POLARITY_DETECTOR** classifier. The conjecture is that the poor results of the alternate approach is because of the absence of a differentiable reasoning function.

Chapter 5

CONCLUSION

5.1 Summary

There is a paradigm shift in the field of deep learning towards developing models and approaches that are interpretable from the perspective of the reasoning process employed. This thesis makes an attempt towards this direction. It mainly focused on creating a new paradigm to solve qualitative question-answering datasets by working on the QuaRTz (Tafjord *et al.*, 2019) and QuaRel (Tafjord *et al.*, 2018) datasets. The model developed here is based on current state-of-the-art deep learning approaches such as BERT (Devlin *et al.*, 2018). In addition to making the approach trainable end-to-end through neural networks, the model constructed here is also interpretable in terms of the reasoning process. Chapter 3 outlined how the model achieves this by defining the specification of the model with respect to problem specific attributes and by performing the required reasoning using these attributes in vector space to compute the correct answer. The model also enforces a set of domain constraints through auxiliary loss functions to achieve interpretability. Chapter 4 outlined the accuracies of the approach formulated here. The model acheives comparable accuracy as the baseline for the QuaRTz dataset, whereas it achieves an improvement in accuracy for the QuaRel dataset. Chapter 4 also exhibited the explainability of the model in terms of the attention vectors of the attributes. It also displayed how the model is capable of variants of qualitative word problems with the directions words flipped with antonyms and negative comparatives.

5.2 Limitations and Future Work

An important point to note in this work is that a problem Q_i is solved using K_i as context. Although some statements like "Reduction in greenhouse gases decreases the levels of ocean acidity" might not necessarily be true in reality, this work infers this statement to be true based on the qualitative knowledge - *An increase in greenhouse gases will increase the changes that are already being seen including in ocean acidity*, as the task of solving problems in the two datasets mainly involves inferring the effect on one concept on another in a qualitative manner.

Since the system developed in this work is modeled based on the two datasets used, there are many assumptions made in the modeling with respect to the structure of problems in the datasets. Some of them are:

- The number of answer options are always 2.
- For problems with different directions as answer options, one option always denotes **MORE** of the query concept, whereas the other option denotes **LESS** of the query concept. Consequently, neither of the answer options can be neutral for such problems.

Some of the ways this work can be extended to answer a broader class of question answering problems are:

- Be able to handle a varying number of answer options.
- Be able to handle problems with higher, lower and neutral intensities.

This work can be extended to address the error cases present in the Results chapter. The reasoning process employed is specifically to reason using qualitative intensifier words like *more*, *closer*, *farther*. It does not deal specifically with problems that contain numbers as intensifiers, or problems that require commonsense to predict the correct answer.

In addition, most of the error cases presented are mainly due to qualitative intensifiers that are ambiguous. This can be resolved by reasoning using a combination of concepts and their intensifiers. The concepts were not used in this work due to lack of reliable annotations. This can be resolved with correct annotations or other techniques like using phrasal embeddings from sources like ConceptNet (Speer *et al.*, 2016) to ground natural language representations of concepts to perform reasoning.

REFERENCES

Devlin, J., M.-W. Chang, K. Lee and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding", (2018).

Gebser, M., R. Kaminski, B. Kaufmann and T. Schaub, "Clingo = ASP + control: Preliminary report", CoRR (2014).

Krishnamurthy, J., P. Dasigi and M. Gardner, "Neural semantic parsing with type constraints for semi-structured tables", in "Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing", pp. 1516–1526 (Association for Computational Linguistics, Copenhagen, Denmark, 2017).

Lai, G., Q. Xie, H. Liu, Y. Yang and E. Hovy, "Race: Large-scale reading comprehension dataset from examinations", (2017).

Lifschitz, V., "What is answer set programming?", in "Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 3", AAAI'08, p. 1594–1597 (AAAI Press, 2008).

Mitra, A., P. Clark, O. Tafjord and C. Baral, "Declarative question answering over knowledge bases containing natural language text with answer set programming", (2019).

Rajpurkar, P., J. Zhang, K. Lopyrev and P. Liang, "Squad: 100,000+ questions for machine comprehension of text", (2016).

Soares, M. A. C. and F. S. Parreiras, "A literature review on question answering techniques, paradigms and systems", Journal of King Saud University - Computer and Information Sciences (2018).

Speer, R., J. Chin and C. Havasi, "Conceptnet 5.5: An open multilingual graph of general knowledge", (2016).

Stewart, R. and S. Ermon, "Label-free supervision of neural networks with physics and domain knowledge", (2016).

Tafjord, O., P. Clark, M. Gardner, W. tau Yih and A. Sabharwal, "Quarel: A dataset and models for answering questions about qualitative relationships", (2018).

Tafjord, O., M. Gardner, K. Lin and P. Clark, "Quartz: An open-domain dataset of qualitative relationship questions", (2019).

Tannen, V., *First-Order Logic: Semantics*, pp. 1138–1139 (Springer US, Boston, MA, 2009), URL `https://doi.org/10.1007/978-0-387-39940-9`$_1$000.

Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser and I. Polosukhin, "Attention is all you need", (2017).

Wolf, T., L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz and J. Brew, “Huggingface’s transformers: State-of-the-art natural language processing”, ArXiv (2019).

Xu, J., Z. Zhang, T. Friedman, Y. Liang and G. V. den Broeck, “A semantic loss function for deep learning with symbolic knowledge”, (2017).

Zellers, R., Y. Bisk, R. Schwartz and Y. Choi, “Swag: A large-scale adversarial dataset for grounded commonsense inference”, (2018).

APPENDIX A

EARLIER APPROACHES

This appendix outlines some of the approaches that were attempted and did not yield satisfactory results in terms of test accuracy and interpretability.

One of the earlier methods that were attempted in this work included reasoning based on soft logic operators. The embedded representations of the *given_world*, *claim_a_world* and *claim_b_world* attributes were computed based on the individual attention vectors and a masked vectors of noun phrases from the question stem. This computation was based on the assumption that the 3 world attributes are often noun phrases in the question stem and/or answer options. A 2d vector NP_p of dimension $number\_of\_noun\_phrases$ * $sequence\_length$ of 0s and 1s represents the noun phrases for a given problem **P**. Each element of the vector represents a noun phrase and contains 1s at the positions corresponding to the tokens that represent that noun phrase. The representations of all the other attributes were similar to those described in Chapter 3 of this document.

The predicted answer was computed based on the truth value for a each answer option, which was computed based on alignment between the two value attributes from the knowledge passage ($value1$, $value2$), the alignment between the *given_value* and the value of the option (*claim_a_value* or *claim_b_value*), and lastly between *given_world* and the worlds of the option (*claim_a_world* and *claim_b_world*). The formula used to compute the truth value is specified below.

$truth_{claim} = OR($

$AND(isProportional, matchedWorld_{(given,claim)}, matchedValue_{(given,claim)}),$

$AND(isProportional, not\ matchedWorld_{(given,claim)}, not\ matchedValue_{(given,claim)}),$

$AND(not\ isProportional, not\ matchedWorld_{(given,claim)}, matchedValue_{(given,claim)}),$

$AND(not\ isProportional, matchedWorld_{(given,claim)}, not\ matchedValue_{(given,claim)}))$

where,

$iSProportional = dotSimilarity(value1, value2)$ denotes the alignment between the two values from the knowledge passage.

The AND, OR and NOT operations were approximated using the *, max and 1-p respectively, where p denotes a vector.

The intuition behind this definition of the truth value was to compute the answer of a problem based on the logical structure that it adhered to. The first two AND gates consider the cases where the two values of the knowledge passage are proportional. If these two values are proportional, the expectation is that both the $claim\_value$ and $claim\_world$ attributes either match $given\_value$ and $given\_world$ as in the first expression, or both $claim\_value$ and $claim\_world$ are opposite to their given counterparts. Whereas, the third and fourth AND gates consider the cases where the two knowledge passage values are inversely proportional. In these two cases, either $claim\_value$ or $claim\_world$ match their given counterparts, but both do not. Ultimately, the OR gate picks the maximum score of these four possibilities as the truth value for a given answer choice.

With this as the model definition, the model did not converge during training. It was observed that using parameter-less soft logical operations did not allow the model to learn this particular end-to-end loss operation.