Deeply Embedded Knowledge Representation & Reasoning For Natural Language Question Answering: A Practitioner's Perspective

Arindam Mitra[1] and **Sanjay Narayana**[2] and **Chitta Baral**[2]
[1]Microsoft
[2]Arizona State University
arindam.mitra@microsoft.com, {snaray48, chitta}@asu.edu

Abstract

Successful application of Knowledge Representation and Reasoning (KR) in Natural Language Understanding (NLU) is largely limited by the availability of a robust and general purpose natural language parser. Even though several projects have been launched in the pursuit of developing a universal meaning representation language, the existence of an accurate universal parser is far from reality. This has severely limited the application of knowledge representation and reasoning (KR) in the field of NLP and also prevented a proper evaluation of KR based NLU systems.

Our goal is to build KR based systems for Natural Language Understanding without relying on a parser. Towards this we propose a method named Deeply Embedded Knowledge Representation & Reasoning (DeepEKR) where we replace the parser by a neural network, soften the symbolic representation so that a deterministic mapping exists between the parser neural network and the interpretable logical form, and finally replace the symbolic solver by an equivalent neural network, so the model can be trained end-to-end.

We evaluate our method with respect to the task of Qualitative Word Problem Solving on the two available datasets (QuaRTz and QuaRel). Our system achieves same accuracy as that of the state-of-the-art accuracy on QuaRTz, outperforms the state-of-the-art on QuaRel and severely outperforms a traditional KR based system. The results show that the bias introduced by a KR solution does not prevent it from doing a better job at the end task. Moreover, our method is interpretable due to the bias introduced by the KR approach.

1 Introduction

Developing agents that understand natural language is a long standing challenge in AI. Towards this, several question answering challenges have been proposed, namely SQuAD (Rajpurkar et al., 2016) containing reading comprehension problems, OBQA (Mihaylov et al., 2018), QASC (Khot et al., 2019) containing science questions requiring inference over multiple facts, ProPara (Mishra et al., 2018), SocialIQA (Sap et al., 2019), RecipeQA (Yagcioglu et al., 2018) requiring understanding of events and effects, QuaRTz (Tafjord et al., 2019b), QuaRel (Tafjord et al., 2019a) requiring qualitative reasoning and bAbI (Weston et al., 2015) containing a broad set of synthetic tasks.

For most of these challenges there exists a KR based methodology which typically says, if "the problem and the associated knowledge is represented as 'R', then there exists an algorithm 'A' which can compute the answer". However, almost no end-to-end system that executes such a solution exists (except for bAbI and QuaRel), as obtaining the desired representation 'R' with precision is a challenging task. For the dataset bAbI, which contains synthetically generated simple sentences, existing semantic parsers work well and thus several KR systems (Mitra and Baral, 2016; Chabierski et al., 2017; Wu et al., 2018) have been implemented for it. But for other datasets, researchers have had to build their own semantic parser when implementing a KR solution. For e.g., the work in (Tafjord et al., 2019a) has developed the QuaSP$^+$ translation system for QuaRel. Data collection for training a semantic parser is a costly process and often parser error becomes a bottleneck to the final system performance. Our goal is eliminate reliance on a semantic parser and to allow rapid implementation of KR based solutions so that the gap between "there is a KR solution" and "there is a system implementing a KR solution" diminishes.

Roughly speaking, our proposed approach takes a KR solution and simulates it in a Neural Network. There are three design choices that are involved in the construction of the simulator Neural Net-

Proceedings of 4th Workshop on Structured Prediction for NLP, pages 102–111
November 20, 2020. ©2020 Association for Computational Linguistics

work. The first design process aims to answer the following question: “How to encode the symbolic representation ‘R’ in terms of vectors so that a deterministic process can convert the vectors back to the original symbolic form?”. The second design process aims to construct a neural network which is responsible for computing the desired vector encoding of ‘R’. The third process, implements the reasoning algorithm ‘A’ in a neural network which takes as input the vector encoding of the symbolic representation ‘R’. The parameters of the networks are learned jointly in an end-to-end fashion. We call this approach, Deeply Embedded Knowledge Representation & Reasoning (DeepEKR).

In this work, we describe a DeepEKR solution for the task of qualitative problem solving (Table 1). We describe a standard KR solution and then describe a way to encode it in a Neural Network. The resulting system is evaluated on the two available datasets, namely Quarel and Quartz. In our evaluation we seek the answer to the following two questions: 1) Can the DeepEKR system outperform the available KR baseline? We find the answer to be yes. 2) Can the DeepEKR system outperform the state-of-the-art? We find the answer to be yes for the QuaRel dataset, for the QuaRTz dataset the performance is same as that of the existing state-of-the-art system. The main contributions of our work is that we propose a novel method to implement a KR solution without relying on a natural language parser and provide a proof of concept towards that.

2 Qualitative Word Problem Solving

A noticeable portion of textual knowledge, particularly in science, economics, and medicine, are qualitative in nature, i.e. they describe how changing one entity (e.g., diesel car) affects another (e.g., air pollution). To help NLU systems become better at understanding such sentences, recently two datasets, Quarel and Quartz, containing Qualitative Word Problems (Table 1) have been developed. Each qualitative word problem is a multiple choice question (Table 1) and is accompanied by a sentence containing necessary qualitative knowledge, both of which are given as input. The hope is that if the system correctly answers the question, it most likely understands the accompanied knowledge.

3 A KR Solution

A KR solution typically describes a high level language where a parser translates the natural language input and a set of rules which then computes the answer given the translated input.

K_1	Bigger stars produce more energy, so their surfaces are hotter.
Q_1	Jan is comparing stars, specifically a small star and the larger Sun. Given the size of each, Jan can tell that the Sun puts out heat that is (A) greater (B) lesser
K_2	An object with greater mass or greater velocity has more kinetic energy.
Q_2	Milo threw both a basketball and a baseball through the air. if the basketball has more mass then the baseball, which ball has more kinetic energy (A) basketball (B) baseball
K_3	A sunscreen with a higher spf protects the skin longer.
Q_3	Billy is wearing sunscreen with a higher spf than Lucy. who will be protected from the sun for longer? (A) Lucy (B) Billy

Table 1: Examples of Qualitative word problems

3.1 Representation

For Qualitative Word Problems, the input contains two parts. One is the qualitative knowledge sentence and another is the multiple choice question. The qualitative knowledge sentence can be compactly represented as a four tuple :

```
(concept 1 value,
 concept 1 description,
 concept 2 value,
 concept 2 description)
```

The “concept 1 value” and “concept 2 value” takes value from the set {“more”,“less”} whereas the concept descriptions are arbitrary. Each tuple basically describes whether “concept 1” and “concept 2” are proportional to each other or inversely proportional to each other. Table 2 shows the the 4-tuple representation of the knowledge sentences for the problems in Table 1.

(more, size of star, more, production of energy)
(more, mass, more, kinetic energy)
(more, spf of sunscreen, more, skin protection)

Table 2: Representation of the knowledge sentences as 4-tuple. We omit a predicate name (e.g., *knowledge*) for brevity.

Each qualitative fact e.g., “Billy is wearing sunscreen with a higher spf”), or a query with option (hereafter, “claim”) such as “who will be protected from the sun for longer? (option) Lucy” can be compactly represented as a 3-tuple :

```
(concept value,
 concept description,
  frame of reference)
```

A 3-tuple either states or claims that some concept (e.g., " spf of sunscreen") attains certain value (e.g. "more") for some reference of frame (e.g., "Billy"). The multiple choice question in the input describes two claims (Claim A and Claim B) one for each answer option A and B and one key fact (hereafter *Fact*) to distinguish the correct claim. Each multiple choice question for the qualitative word problem thus can be represented as a collection of three 3-tuples as shown in Table 3.

Fact	(more, size, sun)
Claim A	(more, heat, sun)
Claim B	(less, heat, sun)
Fact	(more, mass, basketball)
Claim A	(more, kinetic energy, basketball)
Claim B	(more, kinetic energy, baseball)
Fact	(more, spf of sunscreen, Billy)
Claim A	(more, protection, Billy)
Claim B	(more, protection, Lucy)

Table 3: Representation of multiple choice questions

Each qualitative word problem of interest thus can be represented by $4 + 3 \times 3 = 13$ terms. Out of these, the two terms, *Claim A concept description* and *Claim B concept description* always have the same value in the Quarel and Quartz dataset (See Table 3). Thus there are 12 unique terms. We will refer to this set as T. Among these 12 terms, there exist five special terms, namely $\{$*concept 1 value, concept 2 value, Fact Concept Value, Claim A Concept Value, Claim B Concept Value*$\}$ which takes values from the set $\{$"more","less"$\}$. We will refer to this set containing these five special terms as sT.

3.2 Reasoning

The reasoning algorithm is quite straightforward for the qualitative word problems if the input is presented in the desired symbolic representation. To identify the correct answer choice, one can compute and utilize five indicator variables (propositions) as described below.

Let $I_{Rel|K}$ denote an indicator variable which when *true* denotes that according to the knowledge K, the qualitative concepts (e.g., "size of star" and "production of energy") in the word problem P is **proportional** to each other and if *false* then **inversely proportional**. For each answer choice X (where $X \in A, B$), let $I^X_{Rel|F}$ be another indicator variable which denotes if the concept in claim X is proportionally related to the concept in the given Fact or inversely related. Similarly, for each answer choice X (where $X \in A, B$) let $I^X_{Reference|F}$ denote if the frame of reference in the claim X, e.g., "Billy", (Hereafter, *Claim X Ref*) matches with the frame of reference in the given fact (Hereafter, *Fact Ref*) or not. Each of these indicator variables are computed as follows:

$I_{Rel\|K}$	*Concept 1 Value = Concept 2 Value*
$I^X_{Rel\|F}$	*Claim X Concept Value = Fact Concept Value*
$I^X_{Reference\|F}$	*Claim X Ref = Fact Ref*

Table 4: Definition of Indicator variables

The decision function for an answer choice X, $answer(X)$ can then be defined as follows:

$I_{Rel\|K}$	$I^X_{Rel\|F}$	$I^X_{Reference\|F}$	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	T	T	T

Table 5: Decision function: If answer choice X is the correct answer

For the example 3 in Table 1, $I_{Rel|K}$ is *true*, $I^A_{Rel|F}$ is *true*, $I^B_{Rel|F}$ is *true*, $I^A_{Reference|F}$ is *true*, $I^B_{Reference|F}$ is *false*, thus according to Table 5, $answer(A)$ is *true* but $answer(B)$ is *false*.

4 Encoding the Symbolic Representation with Vectors

In this section we describe, how we encode the symbolic representation in terms of vectors. We model each term t whether it is a concept description (e.g., "spf of sunscreen") , a concept value (e.g., "more") or a frame of reference (e.g., "Billy") in terms of two vectors, namely the term surface vector, a^t and the term content vector, v^t. The term surface vector, a^t captures the attention over the natural language input and surrogates for the symbolic description (in our case, phrases like "spf of sunscreen"). The term content vector v^t surrogates for its meaning. For the terms in sT, such as *Concept 1 value*, which take values from a close set, the dimension of the term content vector v^t is equal to the size of that close set, essentially describing a distribution over the members of the set.

In the symbolic form, each qualitative word problem is represented in terms of 12 terms. In its vector form, each problem is thus represented as 12 pair of vectors. Let m be the length of the input sequence tokens (words or sub-words) containing both the knowledge sentence and the multiple choice question (See Figure 1). Each term surface vectors a^t is then a member of the set $[0,1]^n$ (Figure 1). Ideally, we want a^t to be $\in \{0,1\}^n$, however we don't put such an hard constraint to keep the algorithm differentiable and expect that the learned model will exhibit such behavior.

4.1 Encoding Symbolic Reasoning over Vector Space

The decision function for the symbolic representation works with five boolean indicator variables. To work in the continuous space we relax the boolean indicator variables to take any real value in the range of $[-\infty, 0) \cup (0, +\infty]$. If the value of an indicator variable is less than 0, we assume it is *false* and otherwise it is assumed to be *true*. We first obtain a compact formula for the decision function described by the truth table in Table 5. Even though any truth table can be implemented by layers of *and, or* and *not* gates with neural networks, we try to minimize number of such gates to simplify the model. For the truth table in Table 5, the entire truth table can be modelled with two 2-input XNOR gates as follows: $ans(X) = ((I^X_{Rel|F}\ XNOR\ I^X_{Rel|F})\ XNOR\ I^X_{Reference|F})$. Recall that, a 2-input XNOR gate denotes equivalence and has the following truth table:

A	B	A XNOR B
F	F	T
F	T	F
T	F	F
T	T	T

Table 6: Truth table of a 2-input XNOR gate

With our choice of all negative vales as *false* and all positive values as *true*, we use simple multiplication to model the XNOR gate, thus the decision function $ans(X)$, which denotes if X is the correct answer, takes the following simplified form in the continuous space:
$anser(X) = I^X_{Rel|F} \times I^X_{Rel|F} \times I^X_{Reference|F}$

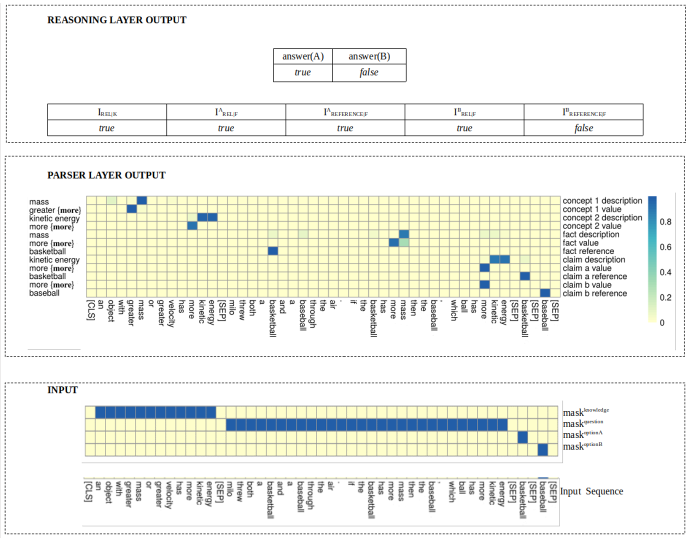

Figure 1: Figure shows a sample input to our model and the predicted output of the parser layer and the reasoning layer. The input masks and the surface term vectors a^t are shown with a heat map over the input sequence. For each of the surface term a^t we also show on the left, the tokens with a weight of more than 0.8. For the five terms in sT we show the value v^t within {} which is "more" for all the five terms for this example.

5 Model

In this section we provide the complete detail about how the term vectors, the indicator variables and the correct answer choice is calculated using the tokenized input containing the qualitative knowledge and the multiple choice question.

Model Input The knowledge sentence (k) and the multiple choice question (*question (A) optionA (B) optionB*) are concatenated together as a single sequence "[CLS] *k* [SEP] *question (A) optionA (B) optionB [SEP]*" and is passed to the Model (Figure 1). Let the length of the input sequence be m. The model then additionally takes as input, four binary masks $\in \{0,1\}^m$, namely $mask_{knowledge}$, $mask_{question}$, $mask_{optionA}$ and $mask_{optionB}$, respectively describing which part of the input belongs to knowledge, question, option A and option B. See Figure 1 for example.

Layer 1: Parser Layer The goal of the parser layer is to recognize 12 important term vector pairs from the input sequence $w_1, ..., w_m$. Towards that, the parser layer first obtains contextual embeddings for each of the token w_i using BERT. Let $e_i \in R^d$ be the embedding for w_i. Those vectors are calculated as follows:

$$e_1, ..., e_m = BERT(w_1, ..., w_m)$$

Let E denote a two dimensional embedding matrix $\in R^{d \times m}$ whose i-th column is e_i.

Using the embeddings in E and the binary masks provided in the input, first the term surface vector $a^t \in [0,1]^m$ are computed for each of the 12 terms in T. Let $f^t(e) : R^d \rightarrow R$ be a linear function of the form $W^t e + b^t$. The j-th component of a vector a^t, i.e., $a^t[j]$ is computed as follows:

$$a^t[j] = \frac{exp(f^t(e_j))}{1 + exp(f^t(e_j))} \times mask^t[j]$$

t	$mask^t$
concept 1 value, concept 1 description, concept 2 value, concept 2 description	$mask_{knowledge}$
Fact description, Claim description, Fact Frame of Reference	$mask_{question}$
Claim A value, Claim A Frame of Reference	$mask_{question}$ + $mask_{optionA}$
Claim B value, Claim B Frame of Reference	$mask_{question}$ + $mask_{optionB}$

Table 7: Describes the value of $mask^t$ for each of the 12 terms.

Table 7 provides the value of $mask^t$ for each of the 12 terms. The $mask^t$ restricts the part of the input sequence that can contain the surface form for the associated term. Since the surface form of each of *Concept 1 value, Concept 1 description, Concept 2 value, Concept 2 description*, should contain tokens from *knowledge sentence* part of the input sequence, $mask^t$ for these four terms are set to be the $mask_{knowledge}$. Other values of $mask^t$ are set accordingly.

The term content (v^t) vector for each of the 7 terms in $T \setminus sT$ which do not take values from the closed set {*more, less*} is computed as follows:

$$v^t = \sum_{j=0}^{m} e_j \times a^t[j]$$

For the remaining 5 terms in sT, we employ a linear function $f^{value} : R^d \rightarrow R$ to obtain the mapping to the closed set {*more, less*}. The term content vector, v^t for each of these 5 terms are defined as follows:

$$v^t = f^{value}(\sum_{j=0}^{m} e_j \times a^t[j])$$

If the value of v^t for these 5 terms are less than 0, we assume that it is aligned towards the value *less*, otherwise it is aligned towards the value *more*.

Layer 2: Reasoning Layer The reasoning layer takes the output from the parser layer and outputs 0 if the correct answer choice is A otherwise it outputs 1. To compute the correct answer, it first obtains the values of the five indicator variables. It computes the value of $I_{Rel|K}, I^A_{Rel|F}, I^B_{Rel|F}$ as follows:

$$I_{Rel|K} = v^{concept\ 1\ value} * v^{concept\ 2\ value}$$

$$I^A_{Rel|F} = v^{fact\ concept\ value} * v^{claim\ a\ concept\ value}$$

$$I^B_{Rel|F} = v^{fact\ concept\ value} * v^{claim\ b\ concept\ value}$$

Recall that each of $I_{Rel|K}, I^A_{Rel|F}, I^B_{Rel|F}$ denotes if a pair of qualitative values are same or not (see Table 4 for definition). With our interpretation of negative meaning *false* and positive denoting *true*, multiplication operator is employed to detect equality.

The value of $I^A_{Reference|F}$ is always set to 1 as we assume the terms in the fact tuple should be translated with respect to the frame of reference in claim A to have an unique translation. Then the value of $I^B_{Reference|F}$ is *true* if the frame of reference in claim A matches the frame of reference in claim B and *false* otherwise. We compute the value of $I^B_{Reference|F}$ as follows:

$$1 - sum^m_{j=0}|a^{claim-a-ref}[j] - a^{claim-b-ref}[j]|$$

Note that we use term surface vector to detect equality. If the two terms (roughly) attends to same positions which should be the case when claim a frame of reference and claim b frame of reference are same (see examples in Table 1 for clarity), the value of $I^B_{Reference|F}$ is positive and thus interpreted as *true*. When the two surface term vectors are disjoint, the value of $I^B_{Reference|F}$ is -1 as $sum^m_{j=0}a^{claim-a-ref}[j]$ = $sum^m_{j=0}a^{claim-b-ref}[j]$ = 1. and is interpreted as *false*.

The score for option A and B is computed as follows,

$$answer(A) = I^X_{Rel|F} \times I^A_{Rel|F} \times I^A_{Reference|F}$$

$$answer(B) = I^X_{Rel|F} \times I^B_{Rel|F} \times I^B_{Reference|F}$$

The answer is 0 if $answer(A) > answer(B)$ other the answer is 1. See Figure 1 for the trace of the reasoning process for the problem 2 in Table 1.

6 Training

Both the Quartz and Quarel dataset provide the correct answer choice for each qualitative word problem. The Quartz dataset additionally provides the concept description (i.e. a^t) and concept value (v^t) annotation for the five terms in sT which we use as additional supervision. This additional information is not supplied for all the word problems in the training dataset. 2280 number of problems out of 2696 problems in the training dataset contain this annotation. The Quarel dataset provides annotation for the concept value for the terms in sT.

In this section we describe our loss function which uses these supervisions and some additional constraints. The loss functions takes as input the following information:

1. $y \in R^2$ contains the confidence score for answer choice A and answer choice B, i.e., $\hat{y} = [answer(A), answer(B)]$.
2. $c \in \{0, 1\}$ which denotes the correct answer.
3. $\hat{v}^t \in \{-1, 1\}$ for the the qualitative values.
4. $\gamma^t \in \{0, 1\}$ which denotes whether the loss function should use the annotation $\hat{v}^t$. This helps to deal with the missing annotation scenario and also in performing some ablation studies.
5. $\hat{a}^t \in \{0, 1\}^m$ for the target value of a^t .
6. $\lambda^t \in \{0, 1\}$ which denotes whether the loss function should use the annotation $\hat{a}^t$.

The loss value L is then computed as follows:

$$\begin{aligned} L = {} & loss^{answer}(y, c) \\ & + \sum_{t \in Cl} \gamma^t * loss^{content}(v^t, \hat{v}^t) \\ & + \sum_{t \in Cl} {}^t * loss^{surface}(a^t, \hat{a}^t) \\ & + loss^{constraint^1} \\ & + loss^{constraint^2} \end{aligned} \tag{1}$$

We use the standard cross entropy function as $loss^{answer}(y, c)$, L1 loss for $loss^{content}$ i.e., $loss^{content}(v^t, \hat{v}^t) = |v^t - \hat{v}^t|$ and binary cross entropy loss function for $loss^{surface}(a^t, \hat{a}^t)$.

The $loss^{constraint^1}$ tells the model that the $a^{concept\ 1\ value}$ and $a^{concept\ 2\ value}$ should be disjoint and similarly $a^{concept\ 1\ description}$ and $a^{concept\ 2\ description}$ should be disjoint. This is computed as follows:

$$\begin{aligned} & loss^{constraint^1} = \\ & mean(a^{concept\ 1\ value} \circ a^{concept\ 2\ value}) + \\ & mean(a^{concept\ 1\ description} \circ a^{concept\ 2\ description}) \end{aligned}$$

Here, $\circ$ denotes element-wise multiplication, $mean(x) : R^m \rightarrow R$ computes the average of all the elements of the input vector x.

Recall that the two options in the multiple choice question either contain two different concept values or two different frame of references. Using this information we add constraints over the term surface vector $a^{claim\ a\ ref}$ and $a^{claim\ b\ ref}$. Let, β if 1 denote that the option choices contain two different frame of reference and 0 otherwise. Note that β can be computed by using the masks $\hat{a}^t$. The $loss^{constraint^2}$ is then computed as follows:

$$\begin{aligned} & loss^{constraint^2} = \\ & \beta * subset(a^{claim\ a\ ref}, mask_{optionA}) + \\ & \beta * subset(a^{claim\ b\ ref}, mask_{optionB}) + \\ & (1 - \beta) * subset(a^{claim\ a\ ref}, mask_{question}) + \\ & (1 - \beta) * *subset(a^{claim\ b\ ref}, mask_{question}) + \\ & (1 - \beta) * mean(|a^{claim\ a\ ref} - a^{claim\ b\ ref}|) \end{aligned}$$

The $subset(a, b)$ function returns 0 if the surface vector a is "subset" of the binary mask b and a positive value otherwise and is defined as follows:

$$subset(a, b) = sum((1 - b) \circ a)$$

Here, $sum(x) : R^m \rightarrow R$ computes the sum of all the elements of the input vector x.

7 Related Work

Our work is related to all the works in Neuro-Symbolic reasoning (Serafini and Garcez, 2016; Cohen et al., 2020; Rocktäschel and Riedel, 2017; Kazemi and Poole, 2018; Aspis et al., 2018; Ebrahimi et al., 2018; Evans and Grefenstette, 2018) that aims at implementing a symbolic theorem prover with Neural Networks. These works provides proof that more complicated symbolic reasoning algorithms than the one used in this work, can be implemented using neural nets. However the algorithms proposed in these work operates over symbolic input, which again calls for a parser. On the other hand several neural systems have been developed for constituency parsing (Stern et al., 2017; Shen et al., 2018), dependency parsing (Chen and Manning, 2014; Dyer et al., 2015), Semantic Role

Constraints	Test Acc %	Concept 1 Value	Concept 2 Value	Fact Concept Value	Claim A Value	Claim B Value
$loss^{answer}$	50	80	82	50	37	60
$loss^{answer}, loss^{constraint^2}$	50	78	82	49	37	60
$loss^{answer}, loss^{constraint^1}$	50	19	18	50	62	39
$loss^{answer}, loss^{surface}$	74.1	16	17	50	50	50
$loss^{answer}, loss^{constraint^1}, loss^{constraint^2}$	50	80	82	50	62	39
$loss^{answer}, loss^{constraint^1}, loss^{constraint^2}, loss^{content}$	50	80	88	50	62	39
$loss^{answer}, loss^{constraint^1}, loss^{constraint^2}, loss^{content}, loss^{surface}$	**79.84**	89	92	80	94	95
$loss^{answer}, loss^{surface}, loss^{content}$	78.18	91	88	78	91	94

Table 8: Ablation Analysis of different supervisions

Labelling (He et al., 2018), parsing to the language of Abstract Meaning Representation (Konstas et al., 2017) or task specific semantic parsing(Dong and Lapata, 2018; Krishnamurthy et al., 2017). These works also provide useful knowledge while constructing a DeepEKR solution.

In this work, the input problem is translated to a set of fixed number of terms. However, depending on the end application the representation format could be a graph, stack, table. Thus the work in Graph Neural Networks (Scarselli et al., 2008; Lamb et al., 2020), which operates over graphs or the Neural State Machine (Hudson and Manning, 2019) that operates over automata is also related to our work.

In this work we have proposed to replace the symbolic representation by vectors so that dependency over an accurate parser can be avoided. With a similar goal, the work in (Mitra et al., 2019b) proposes to use textual entailment to replace the parser. The central idea behind the proposal is, if the input is supposed to be translated to a predicate e.g., *claimA("protection","more","Billy")*, instead of asking the parser to translate it to the symbolic form, generate a textual description for the predicate e.g., "protection is more for Billy" and use a textual entailment system to check if the input string entails it. A drawback of this approach is that generation of the textual description of a symbolic term currently requires handwritten templates. A system, namely **gvQPS** (Mitra et al., 2019a) following this approach has been built for the QuaRel dataset.

Our work is directly related to the QUASP$^+$ system (Tafjord et al., 2019a) for QuaRel that trains a parser to obtain a symbolic representation of a qualitative word problem and uses a symbolic reasoner implemented in Prolog to obtain the answer. Our work is also related to the BERT (Devlin et al., 2018) based multiple choice question solver that takes as input "[CLS] knowledge [SEP] question [SEP] option X [SEP]" and computes the score for option X.

8 Experiments

We evaluate our system on the QuaRTz and QuaRel dataset. The QuaRTz dataset contains a total of 3864 problems. The train, dev and test split respectively contain 2696, 384 and 784 problems. The QuaRel dataset contains a total of 2771 problems. The train, dev and test split respectively contain 1941, 278 and 552 problems. We have used the *bert-large-uncased-whole-word-masking* model in our experimentation.

Performance on QuaRTz Table 9 compares the accuracy of our system (DeepEKR) with the two reported solvers, namely BERT (standard BERT multiple choice question solver trained on the QuaRTz dataset) and BERT-PFT-Race (BERT multiple choice question solver trained on the Race dataset (Lai et al., 2017) and then on the QuaRTz dataset) . Our system achieves same accuracy to that of the BERT-PFT-Race model. However, DeepEKR provides better interpretability.

Models ↓	Test Acc.
BERT	67.7
BERT-PFT-Race	79.8
DeepEKR	79.8

Table 9: Performance of various models on QuaRTz

Ablation Analysis on Supervision The loss function takes five different supervisions as described in equation 1. Table 8 displays the effect of different combination of supervisions on the question-answering accuracy on the test set and the accuracy of $v^t \in \{$“less”,“more”$\}$ for the $t \in sT$. We observe that a combination of all constraints results in the best test accuracy. However, $loss^{surface}$ i.e. the supervision for term surface vector is the most significant one, as without this supervision accuracy remains stuck at 50%. Due to this, while training on QuaRel, we either pretrain the model on QuaRTz or expand the QuaRel training data with QuaRTz training data.

Performance on QuaRel Table 10 compares the accuracy of our system on the QuaRel dataset. DeepEKR model first trained on QuaRTz and then later fine-tuned on QuaRel achieves the state-of-the-art-accuracy.

Models ↓	Test Acc.
BERT	53
BERT-PFT-Race	79.89
BERT-PFT-QuaRTz	53
BERT-PFT-Race and PFT-QuarTz	77
QuaSP+	68.7
gvQPS	76.63
DeepEKR PFT on QuaRTz	**81.15**
DeepEKR augmented with QuaRTz training data	78.98

Table 10: Performance of various models on Quarel

8.1 Error Analysis

We carefully examine all the 87 examples in the dev set of the QuaRTz dataset where the system picks the incorrect answer. We break down the errors in 5 categories.

Incorrect Value Prediction The majority of the errors (41) fall in this category where a^t is correctly computed for the terms t in sT but one of $I_{Rel|K}$ or $I^X_{Rel|F}$ is wrong. Table 11 displays an example with this error. Here, the two concepts being compared are *energy of vibrations* and *proximity of particles*. Our system incorrectly classifies $v^{fact\ concept\ value}$ (“further”) as “more” even though the associated concept is *proximity of particles* resulting in an error in the computation. This happens as “farther” often correlates with “more” in the dataset. We believe adding more examples to teach the model that $v^{fact\ concept\ value}$ sometimes depends on concept description is necessary to deal with this issue.

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? (A) decrease (B) increase

Table 11: An Example of Incorrect Value Prediction

Attention over Incorrect Tokens For 28 problems, the incorrect token gets a high attention score i.e. a^t is wrong, leading to incorrect v^t and ultimately in an incorrect prediction. This occurs for the example in Table 12, where $a^{fact\ concept\ value}$ points to the token “increases” but does not contain “removing” which results in incorrect v^t.

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? (A) decrease (B) increase

Table 12: An example of Attention over Incorrect Tokens.

Others For the reaming 18 problems, 9 requires numerical reasoning (number comparisons), 4 requires commonsense knowledge such as “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?(A) Mercury (B) Mars”. For 5 problems the gold answer provided is actually wrong and the model actually predicted the correct answer.

9 Conclusion

Knowledge Representation and Reasoning (KR) based solutions are interesting for Natural Language Understanding as they are interpretable and can work with declarative knowledge. However, systems that implement KR solution with traditional parser and symbolic solvers normally fall short on performance when compared to neural systems. These observations and issues related to parser and symbolic reasoning have resulted in less interest towards KR solutions. However, we show that we can take a KR solution and implement in a way that is competitive with neural systems and is also explainable. For the qualitative word problems, the reasoning is fairly simple. Our future work includes applying this method to other areas requiring more complex reasoning.

References

Yaniv Aspis, Krysia Broda, and Alessandra Russo. 2018. Tensor-based abduction in horn propositional programs. CEUR Workshop Proceedings.

Piotr Chabierski, Alessandra Russo, and Mark Law. 2017. Logic-based approach to machine comprehension of text.

Danqi Chen and Christopher D Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 740–750.

William Cohen, Fan Yang, and Kathryn Rivard Mazaitis. 2020. Tensorlog: A probabilistic database implemented using deep-learning infrastructure. *Journal of Artificial Intelligence Research*, 67:285–325.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Li Dong and Mirella Lapata. 2018. Coarse-to-fine decoding for neural semantic parsing. *arXiv preprint arXiv:1805.04793*.

Chris Dyer, Miguel Ballesteros, Wang Ling, Austin Matthews, and Noah A Smith. 2015. Transition-based dependency parsing with stack long short-term memory. *arXiv preprint arXiv:1505.08075*.

Monireh Ebrahimi, Md Kamruzzaman Sarker, Federico Bianchi, Ning Xie, Derek Doran, and Pascal Hitzler. 2018. Reasoning over rdf knowledge bases using deep learning. *arXiv preprint arXiv:1811.04132*.

Richard Evans and Edward Grefenstette. 2018. Learning explanatory rules from noisy data. *Journal of Artificial Intelligence Research*, 61:1–64.

Luheng He, Kenton Lee, Omer Levy, and Luke Zettlemoyer. 2018. Jointly predicting predicates and arguments in neural semantic role labeling. *arXiv preprint arXiv:1805.04787*.

Drew Hudson and Christopher D Manning. 2019. Learning by abstraction: The neural state machine. In *Advances in Neural Information Processing Systems*, pages 5901–5914.

Seyed Mehran Kazemi and David Poole. 2018. Relnn: A deep neural model for relational learning. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Tushar Khot, Peter Clark, Michal Guerquin, Peter Jansen, and Ashish Sabharwal. 2019. Qasc: A dataset for question answering via sentence composition. *arXiv preprint arXiv:1910.11473*.

Ioannis Konstas, Srinivasan Iyer, Mark Yatskar, Yejin Choi, and Luke Zettlemoyer. 2017. Neural amr: Sequence-to-sequence models for parsing and generation. *arXiv preprint arXiv:1704.08381*.

Jayant Krishnamurthy, Pradeep Dasigi, and Matt Gardner. 2017. Neural semantic parsing with type constraints for semi-structured tables. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1516–1526.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. Race: Large-scale reading comprehension dataset from examinations. *arXiv preprint arXiv:1704.04683*.

Luis Lamb, Artur Garcez, Marco Gori, Marcelo Prates, Pedro Avelar, and Moshe Vardi. 2020. Graph neural networks meet neural-symbolic computing: A survey and perspective. *arXiv preprint arXiv:2003.00330*.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. *arXiv preprint arXiv:1809.02789*.

Bhavana Dalvi Mishra, Lifu Huang, Niket Tandon, Wen-tau Yih, and Peter Clark. 2018. Tracking state changes in procedural text: a challenge dataset and models for process paragraph comprehension. *arXiv preprint arXiv:1805.06975*.

Arindam Mitra and Chitta Baral. 2016. Addressing a question answering challenge by combining statistical methods with inductive rule learning and reasoning. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Arindam Mitra, Chitta Baral, Aurgho Bhattacharjee, and Ishan Shrivastava. 2019a. A generate-validate approach to answering questions about qualitative relationships. *arXiv preprint arXiv:1908.03645*.

Arindam Mitra, Peter Clark, Oyvind Tafjord, and Chitta Baral. 2019b. Declarative question answering over knowledge bases containing natural language text with answer set programming. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3003–3010.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Tim Rocktäschel and Sebastian Riedel. 2017. End-to-end differentiable proving. In *Advances in Neural Information Processing Systems*, pages 3788–3800.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. 2019. Social iqa: Commonsense reasoning about social interactions. In *EMNLP 2019*.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. 2008. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80.

Luciano Serafini and Artur S d'Avila Garcez. 2016. Learning and reasoning with logic tensor networks. In *Conference of the Italian Association for Artificial Intelligence*, pages 334–348. Springer.

Yikang Shen, Zhouhan Lin, Athul Paul Jacob, Alessandro Sordoni, Aaron Courville, and Yoshua Bengio. 2018. Straight to the tree: Constituency parsing with neural syntactic distance. *arXiv preprint arXiv:1806.04168*.

Mitchell Stern, Jacob Andreas, and Dan Klein. 2017. A minimal span-based neural constituency parser. *arXiv preprint arXiv:1705.03919*.

Oyvind Tafjord, Peter Clark, Matt Gardner, Wen-tau Yih, and Ashish Sabharwal. 2019a. Quarel: A dataset and models for answering questions about qualitative relationships. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7063–7071.

Oyvind Tafjord, Matt Gardner, Kevin Lin, and Peter Clark. 2019b. Quartz: An open-domain dataset of qualitative relationship questions. *ArXiv*, abs/1909.03553.

Jason Weston, Antoine Bordes, Sumit Chopra, Alexander M Rush, Bart van Merriënboer, Armand Joulin, and Tomas Mikolov. 2015. Towards ai-complete question answering: A set of prerequisite toy tasks. *arXiv preprint arXiv:1502.05698*.

Benjamin Wu, Alessandra Russo, Mark Law, and Katsumi Inoue. 2018. Learning commonsense knowledge through interactive dialogue. In *Technical Communications of the 34th International Conference on Logic Programming (ICLP 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

Semih Yagcioglu, Aykut Erdem, Erkut Erdem, and Nazli Ikizler-Cinbis. 2018. Recipeqa: A challenge dataset for multimodal comprehension of cooking recipes. *arXiv preprint arXiv:1809.00812*.