A Neuro-Symbolic Approach for Question Answering on Scholarly Articles

Komal Gupta[1], **Tirthankar Ghosal**[2], **Asif Ekbal**[1]
[1]Department of Computer Science and Engineering,
Indian Institute of Technology Patna, India
(komal_2021cs16, asif)@iitp.ac.in
[2]Institute of Formal and Applied Linguistics,
Faculty of Mathematics and Physics,
Charles University, Czech Republic
ghosal@ufal.mff.cuni.cz

Abstract

The number of research articles is increasing exponentially. It has become difficult for researchers to stay updated with the latest development in science with the deluge of papers. Hence, keeping abreast with the current literature is one of the most significant challenges to present-day researchers. However, if one can query a scientific article, they can quickly comprehend it and elicit the required information. Hence, a question-answering (QA) system on scholarly articles would be a helpful assistant for researchers to survey the literature. Recently logic-infused deep networks have been showing good promise for solving several downstream NLP tasks. Here in this paper, we implement a neural network-based symbolic approach for QA on scholarly articles. We incorporate logical boolean functions into the deep network, significantly improving the model's performance without additional parameters. Further, we reduce the dependency on domain-specific training data by using external knowledge from the ConceptNet. We perform our experiments on the benchmark *ScholarlyRead* dataset and achieve significant performance improvement ($\sim$ double) over the baseline approach. We would make our codebase available here [1].

1 Introduction

The explosive growth in the number of scholarly articles is posing a significant *information overload* to the present-day researcher. Researchers spend a considerable amount of their time finding relevant articles and reading them to find answers to their queries. As we all know literature survey is an essential part of the research life cycle; however, it is gradually becoming impossible to keep abreast with all the knowledge even in a particular domain. With the intrusion of Artificial Intelligence in almost all spheres of life, it makes sense to have an intelligent tool to query the scholarly literature to extract desired answers. If researchers can get answers to some usual and fundamental questions related to the article, they can consume more literature in a given time and hence tackle the *scholarly burden*. With this motivation, we explore the problem of question answering (QA) on scholarly articles. We attempt to develop a scientific article query model to help researchers save time. Given a paragraph and a query, the aim is to select start and end indices from the context (a sequence of the words) that answers the question. A usable QA model should comprehend the scholarly text and extract target information from the discourse upon a user query.

Our method makes use of symbolic logic (Besold et al., 2017) incorporated into neural networks for reasoning (Cingillioglu and Russo, 2018) over the discourse upon a user query. In this paper, we take advantage of the subtle connection between neural networks and logic (Anthony, 2003). We use the domain knowledge articulated as the first-order logic predicates. We directly incorporate the structural knowledge into a deep neural network model (BiDAF (Seo et al., 2016)) without any changes in the training. We use the popular BiDAF (Bidirectional Atten-

[1]https://github.com/92noname/Neuro-Symbolic-QA-Model-on-Scholarly-Articles

tion Flow) model as the baseline here. In this model, query words line up with the context words in the hidden layer for finding the answer indices. We consider augmenting an external resource, ConceptNet, to find relations (such as *synonyms, is-a, distinct*) between the words in our proposed model. Our main contribution in the current work is to build a robust QA system by injecting logical rules into a deep neural framework that achieves state-of-the-art performance for the benchmark *ScholaryRead* dataset (Saikh et al., 2020).

We organize the rest of our paper as follows. Section 2 discusses the related work. Section 3 defines the task. Section 4 and Section 5 explain the problem setup and augmented BiDAF model with logic, respectively. We present the dataset description in Section 6. Section 7 presents our results and error analysis. Analysis and observation are in Section 7.5. Finally, Section 8 concludes with our findings and directions to work further.

2 Related Work

In this section, we focus on some notable works for question answering in NLP. We also perform surveys of some recent works where the infusion of logic in neural models has proven effective for the downstream NLP tasks.

2.1 Question Answering

Question answering from document discourse is a very popular NLP task that simulates machine reading comprehension. Kadlec et al. (2016) used a simple model which directly extracts the answer from the context via the neural attention mechanism. The answer is only one word in the model. Hermann et al. (2015) presents a dynamic attention-based model that performs better than a single fixed query vector to attend on context words on CNN/Daily Mail dataset. Allam and Haggag (2012) provides an overview of QA and its system architecture and the previous related work comparing each research against the others concerning the cover components and the approaches that follow. Usbeck et al. (2019) present a new online benchmarking framework for QA that relies on the FAIR (Findable, Accessible, Interoperable, Re-Usable) principles to support the fine-grained evaluation of QA systems. Gupta et al. (2021) proposed a hierarchical-based deep multi-modal neural network that classifies end-user questions and then incorporates a query-specific approach for answer prediction. Esteva et al. (2020) presented a retriever-ranker semantic search engine designed to handle complex queries over the COVID-19 articles, potentially aiding overburdened health workers in finding scientific answers during a time of crisis. Liakata et al. (2013) presented an approach that exploits automatically generated scientific discourse annotations to create a content model for summarising scientific articles and finally demonstrated the usefulness of the summaries by evaluating them in a complex question answering task. Kolomiyets and Moens (2011) presented the QA task from an information retrieval perspective. It emphasizes the importance of retrieval models, i.e., representations of questions and information documents and retrieval functions used to estimate the relevance between a query and an answer candidate.

2.2 Neural Networks with Logic

Anthony (2003) investigate the bridge between the neural network and the boolean functions. Hu et al. (2016) implements a systematic distillation method that conveys the structural knowledge of logic rules into the neural network's weights. In Chang et al. (2012), the authors present a Constrained Conditional Models (CCMs) framework that augments linear models with declarative constraints as a way to support decisions in an expressive output space while maintaining modularity and tractability of training. In this work, (Srikant and O'Reilly, 2021) reviewed three sets of recent results in human cognition experiments in natural language comprehension, in natural language inference, and computer program comprehension, a field bearing similarities to natural language. In this paper, (Ma et al., 2019) performed a survey of recent commonsense QA methods. They systematically analyzed the popular knowledge resources and knowledge integration methods across benchmarks from multiple commonsense datasets. Liu and Singh (2004) introduced a large-scale knowledge base such as ConceptNet, which we also use in our current work for injecting additional knowledge to our model.

Our work is inspired from Saikh et al. (2020) for QA on scholarly papers, but with an additional infu-

Paragraph:
Being digitized the communication system channels are having a severe workload in the current scenario . A number of increases in digital devices require more address variable . The internet and 4G services available in the global world are able to provide good service to both stationary and moving devices . Huge number of devices can be introduced in a specific region of the network . Devices can be operated in any layer of the network . Most of challenges are coming in cross layer hand - off as delay and protocols available different network are different . For the above kind of handoff problems this paper proposes a technology which is including a variety of header formats and a suitable protocol for cross layer handoff in IPv6 header format . The work is done on a layer wise hand off and dual communication with two base stations for a certain time while hand - off occurs . Multiple header format supports reliable uninterrupted and delay avoidance in the communication system . The security and authentication are enhanced with two control units .

Question: What kind of internet available in the global market ?
Answer: communication system channels

Question: Where are the internet and 4G services available ?
Answer: global world

Question: Where huge number of devices can be introduced ?
Answer: specific region.

Figure 1: **Example of reading comprehension from the ScholarlyRead dataset (Saikh et al., 2020).**

sion of logic with deep neural network models.

3 Task Definition

Given an abstract of a scientific paper, the input to the model is the context $c_1, c_2, c_3, ...c_m$ and the query is $q_1, q_2, q_3 ..... q_n$. The output should be $c_i, c_{i+1}, ...c_k$. where $m >= k$. There can be multiple questions in a given context, as shown in Figure 1. The task is to extract the relevant answers from the text under concern (see the highlighted portions in multiple colors in the above figure).

4 Problem Setup

This section discusses the notations, assumptions, how to incorporate the logic into the neural network, constrained and constrained auxiliary layer, and steps of building an augmented neural network model for the problem at hand.

4.1 Notations and Assumptions

The architecture of the neural network is a kind of directed graph. Let us consider a directed graph G (V, E), where V (nodes) represents the network's neurons, and E (edges) shows the direction of the information process between the two nodes. Let us assume that the directed graph G has two nodes, a and b. If the edge is from a to b, then a is an upstream neuron, and b is a downstream neuron. The semantics of some nodes (extra neurons) will assign with the model design (on the attention layer). We will assume extra neurons are given (such as word relatedness from the ConceptNet). The main objective is to use the declarative rule (logic) to augment the neural

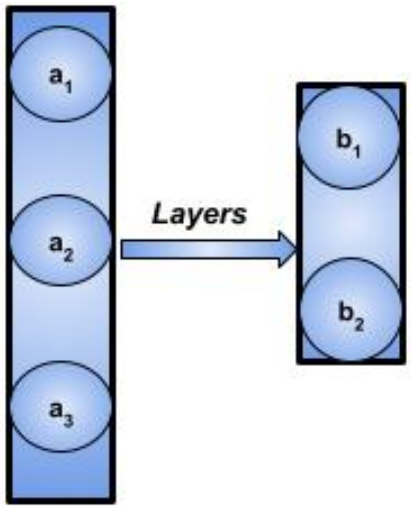

Figure 2: **Examples of the computation graph. Here, a_1, a_2, a_3 represent the neurons of the layer A and b_1, b_2 represent the neurons of the layer B and show the information flow from layer A to layer B.**

network with some extra neurons. In the rest of the paper, uppercase letters (A_i, B_i) denote the predicate graphs associated with the computation graph, and lower case letters (a_i, b_i) represent the nodes in the computation graph.

4.2 Incorporating Soft Logic in Neural Network

Extra edges are incorporated to add logic to the neural network (for augmenting word relation nodes in the neural network) into the computation graph. Before augmenting the computation graph with conditional form, we need to check the cycle in the constraints. Let us define the cyclicity in a conditional statement as:

The statement $A_1 \wedge B_1 \rightarrow A_2 \wedge B_2$ is cyclic with respect to the graph. in contrast, the statement $A_1 \wedge A_2 \rightarrow B_1 \wedge B_2$ is acyclic as shown in Figure 2.

We will focus on the conditional statements of the form $Z \rightarrow Y$. Here, Z is the antecedent that can be conjunction or disjunction of literals, and Y is the consequent consisting of a single literal. The neuron associated with Y is defined as:

$$y = g(Wx) \tag{1}$$

Where g is an activation function, W represents the weight of the network, and x is the input. In order to

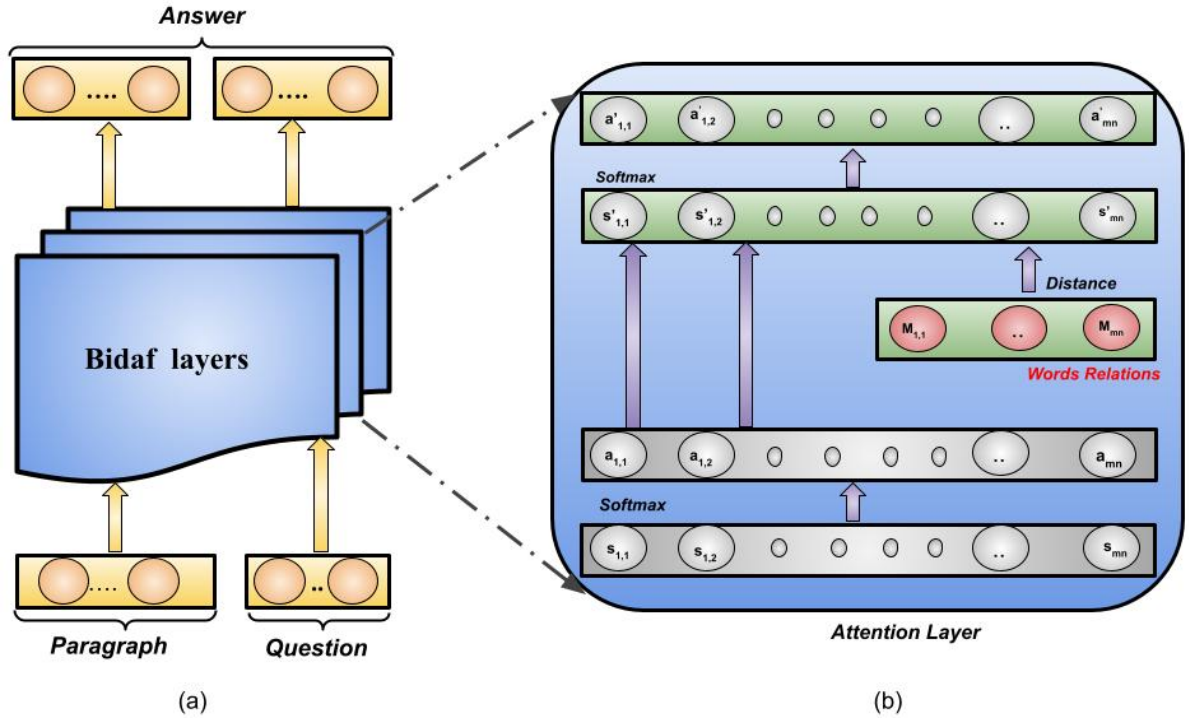

Figure 3: **(a) End-to-end BiDAF model and (b) The computation graph of the attention layer of the Augmented BiDAF model is shown enlarged. Computation graph attention layer using R_2. The red neuron represents the relation of the words. Constrained attention and score are represented a' and s', respectively.**

insert logic into the neural network, some conditions must be satisfied in the conditional statement:

1. Conditional statement must be in Z $\rightarrow$ Y form.
2. In conditional statement Z $\rightarrow$ Y, Z must be in conjunctive/disjunctive form, and Y is single literal.
3. Conditional statement Z $\rightarrow$ Y must be in acyclic form.

4.2.1 Constrained Layer

Let us assume that z be the neuron's vector, and Z is predicate logic, and z is connecting with the predicate logic in Z. With the help of distance function, we can adjust the score of the downstream neurons based on the state of upstream neurons. The constrained neural layer is defined as:

$$y = g(Wx + \rho d(z)) \tag{2}$$

Our objective is to increase the value of y whenever Z is true. Here, d is a distance function, and ρ is an actual value hyperparameter where $\rho \geq 0$.

1. **Distance Function** : The ideal distance function we want is the indicator for statement Z.

$$d_{ideal}(z) = \begin{cases} 1, & \textit{if Z holds}, \\ 0, & \textit{otherwise}. \end{cases} \tag{3}$$

2. **Scaling** : Distance function controls the pre-activation score of downstream neurons. The scaling factor controls the neuron's score. When ρ is positive, the distance function will dominate the downstream neurons. When the value of ρ is negative or significantly less (close to zero), the output depends on both *Wx* and distance function.

4.2.2 Constrained Auxiliary Layers

For the bidirectional constraint Y $\leftrightarrow$ Z, we define a constrained auxiliary layer as:

$$y = d(z) \tag{4}$$

Where d is a distance function, and z and y are the upstream and downstream neurons, respectively. The advantage of this layer is that we can use the same distance function as shown in Table 1.

Logic Operations	Distance function d(z_1, z_2)
z_1 AND z_2	max(0, z_1+z_2-1)
z_1 OR z_2	min(1, z_1+z_2)
NOT z_2	(1-z_1)

Table 1: **Distance Function**

4.3 Building Augmented Neural Network

Here, the steps to build a neural augmentation network from the given conditional statements and computational graphs are described as below:

1. First, ensure that all the antecedents are in conjunctive normal form (CNF) and Disjunctive normal form (DNF).

2. Convert the CNF/DNF antecedents into the distance functions using the constrained Table 1.

3. Using the distance function, build the constrained layer and constrained auxiliary layer by replacing the original layer with the constraint layer.

4. Use the logic augmented neural network for end-to-end training.

5 Methodology

The logic augmented BiDAF model is shown in Figure 3. Figure 3a shows the BiDAF model, and Figure 3b shows the augmentation of the logic rule R_2 (equation 6) to the attention layer of the BiDAF model. The word relatedness layer contains the relation between the query words and the context words. These relations are extracts using the ConceptNet.

The model starts with a context sequence of words P = $p_1, ..., p_n$ and query sequence of words Q = $q_1, ..., q_m$. It returns a continuous sub-sequence of words (span) S = $p_i, ..., p_{i+j}$ as output that represents the best possible answer for the query. We perform the experiments for logic augmented neural network over *ScholarlyRead* dataset and compare the results with the BiDAF model (Seo et al., 2016).

5.1 Augmented BiDAF Model

Augmented BiDAF is a multi-stage hierarchical model. The model has six layers, *viz.* (i). Character Embedding layer, (ii). Word Embedding layer, (iii). Contextual Embedding layer, (iv). Attention Flow layer, (v). Modeling layer, and (vi). Output layer. The input layer is divided into: (a). Character Embedding layer and (b). Word Embedding layer. We briefly describe each layer below:

5.1.1 Input Embedding Layer

This layer is responsible for mapping every word into high dimensional vector. Let $p_1, ...p_t$ and $q_1, ...q_j$ represent the words in the input context paragraph and query, respectively. Character convolutional neural network (CNN) generates character embeddings to handle the out-of-vocabulary (OOV) words. For the word level embedding, we use the pre-trained word vector Glove (Pennington et al., 2014). Character level embedding and word-level embedding are are concatenating and pass to the highway network (Srivastava et al., 2015).

5.1.2 Contextual Embedding Layer

In the contextual layer, we use the bi-directional long short-term memory (Bi-LSTM). The output of the highway network is passed to the bi-directional LSTM networks. In this layer, we utilize contextual cues from the surrounding words to filter the word embedding.

5.1.3 Bi-directional Attention Flow (Attention guide with external source)

In the BiDAF model, Seo et al. (2016) calculate the attention in two directions: *query to context* and *context to query*. In the Augmented BiDAF model, we guide the attention value with the help of a knowledge-driven approach for which we use ConceptNet. The augmented model on attention layer using logic (Section 5.2) is shown in Figure 3.

5.1.4 Modeling Layer

The query-conscious representations of context words are input to the modeling layer. The output of this layer uses to capture the information (words) which interacts among the context words and query words.

5.1.5 Output Layer

The output layer gives the probability of start and end indices. These two indices with the highest probabilities correspond to the answer's position where it starts and ends in the context.

S.No	Error Type	Example
1	**Imprecise answer boundaries**	**Context:** The goal of every network routing protocol is to direct the traffic from source to destination maximizing the network performance. The Ant Colony Optimization ACO based routing protocol is efficient when used to dynamically route traffic. **Question:** What kind of outing protocol is efficient ? **Prediction 1 :** Colony Optimization ACO **Prediction 2 :** Ant Colony Optimization **Answer:** Ant Colony Optimization
2	**Syntactic complications and ambiguities**	**Context:** This results in lowered network efficiency and poor Quality of Service QoS. A number of routing protocols have been developed to deal with network traffic. The goal of every network routing protocol is to direct the traffic from source to destination maximizing the network performance. The Ant Colony Optimization ACO based routing protocol is efficient when used to dynamically route traffic. **Question:** How does The Ant Colony Optimization based routing protocol efficient? **Prediction 1 :** network efficiency **Prediction 2 :** route traffic **Answer:** route traffic
3	**Paraphrase problems**	**Context:** This results in lowered network efficiency and poor Quality of Service QoS. A number of routing protocols have been developed to deal with network traffic. The goal of every network routing protocol is to direct the traffic from source to destination maximizing the network performance. **Question:** What is the goal of every network routing protocol in this paper ? **Prediction 1 :** network efficiency **Prediction 2 :** network performance **Answer:** network performance
4	**External Knowledge**	**Context:** Congestion packet loss and increased response-time due to network traffic are common problems in most networks. This results in lowered network efficiency and poor Quality of Service QoS. A number of routing protocols have been developed to deal with network traffic. The goal of every network routing protocol is to direct the traffic from source to destination maximizing the network performance. The Ant Colony Optimization ACO based routing protocol is efficient when used to dynamically route traffic. **Question:** What does priority scheme improve ? **Prediction 1 :** Congestion packet loss **Prediction 2 :** ACO algorithm **Answer:** ACO algorithm
5	**Incorrect preprocessing**	**Context:** In this paper we propose the possibility for 3G operators to share those community-deployed wireless infrastructures for their 3G backhauling needs. This would permit them to have a low-cost backhaul solution for their rural 3G small cells in those areas where the expected demand of 3G services does not ensure enough revenues to justify the deployment of dedicated infrastructures. **Question:** What kind of need that make 3 G operators to share those community - deployed wireless infrastructures ? **Prediction 1 :** G backhauling **Prediction 2 :** 3G backhauling **Answer:** 3G backhauling

Table 2: **The examples for each category of error using BiDAF in ScholarlyRead dataset which are resolved with BiDAF+R_1 . The baseline model BiDAF results are shown by *prediction 1 (Saikh et al., 2020)* and the augmented model using rule R_1 results are shown by *prediction 2*. Showing the ground truth labels by *answer*.**

5.2 Augmentation Rule

We adopt the augmentation rules from Li and Srikumar (2019). They are using external knowledge source such as ConceptNet to guide the attention neurons. First, we define the following notations:

1. M_{ij}: If the paragraph word p_i related to query word q_j using the ConceptNet edges.
2. A_{ij}: p_i match with q_j based on the unconstrained model decision.
3. A'_{ij}: p_i match with q_j based on the constrained model decision.

Using these notations, we define two rules:
Let C be a set of words in paragraph and query word:

1. According to the rule R_1 that if two words are related so they should be aligned :

$$R_1 : \forall_{i,j} \in C, M_{ij} \rightarrow A'_{ij} \tag{5}$$

2. Rule R_2 align the two related words when the constrained and unconstrained models agree over a similar decision.

$$R_2 : \forall_{i,j} \in C, M_{ij} \wedge A_{ij} \rightarrow A'_{ij} \tag{6}$$

6 Dataset Description

For training our model, we use the ScholarlyRead dataset from Saikh et al. (2020). The dataset is span-of-word-based prediction (Span Prediction) for Machine Reading Comprehension (MRC), where the model has to extract span-of-words as the answer to a query based on the context. Essentially the dataset contains questions with their corresponding answer span from the context. It comprises approximately 300 articles from two Elsevier Computer Science journals, *viz.* Artificial Intelligence (ARTINT) and Computer Networks (COMNET), in which approximately 10K data samples (context, query, answer) are manually annotated. The number of instances in training, validation and the test sets are 8500, 1500, 500, respectively. In this dataset, the authors use the abstract of the research articles as the context because the abstract contains the scientific article's summary. Our analysis shows that possible answers cover 6.5 % proper nouns, 46.5 % common nouns, 17.7 % plural nouns, 21 % adjectives, 3.5 % verbs and 4.8 % others. Kindly refer to the dataset paper for more details on the dataset construction and organization.

7 Experiments, Results and Analysis

We train two versions of the augmented BiDAF model with rules R_1 and R_2. We keep the configuration of the models, hyperparameters identical in both cases.

7.1 Experimental Setup and Evaluation Metrics

We train the models using 8500 samples and validate on 1500 samples. The length of the longest paragraph in the training set is 624 tokens, and the length of the most extended query is 37 tokens. For representation, we use the Glove pre-trained embeddings (Pennington et al., 2014) having dimensions of 300. Along with word embeddings, we also use character embeddings to obtain better representation for out-of-vocabulary (OOV) words. For the character embeddings, we use a convolutional neural network (CNN). We use one dimension filters for CNN character embedding, each with a width of 7. We use a two-layer highway network (Srivastava et al., 2015) for the concatenation of word and character embeddings. Finally, we test the model on 500 instances. We use two evaluation matrices, F1 and Exact Match (EM), to evaluate the model's performance. F1 measures the overlap portion between prediction and ground truth answer while EM finds if the prediction matches exactly the ground truth. If the prediction and the ground truth are the same, the Exact Match score is 1; otherwise, 0. We tokenize each paragraph and query via the Spacy [2] tokenizer. The hidden dimension of the model is 100. The total number of training parameters is 2.6 million. We train the model for 35 epochs with the learning rate and dropout as 0.001 and 0.2, respectively, and use the Adam (Kingman and Ba, 2015) optimizer.

7.2 Baseline

We compare our model with the BiDAF model as a baseline (Seo et al., 2016). The BiDAF basic architecture has six central layers: 1) Character Embedding layer, 2) Word Embedding layer, 3) Contextual Embedding layer, 4) Attention Flow layer, 5) Modeling layer, and 6) Output layer. This model has a fairly standard template, which we also follow in our augmented model architecture.

[2] https://spacy.io/api/tokenizer

Model	F1 Score	Exact Match
BiDAF	37.3	20.6
BiDAF+R1	74.9	61.0
BiDAF+R2	74.6	60.8

Table 3: **Evaluation results on the ScholarlyRead dataset. Each score represents the average span F1 score and Exact Match on our test set.**

Train %	Data	BiDAF+R1	BiDAF+R2
10%	Squad	61.5	60.7
10%	ScholarlyRead	52.6	47.5
100%	Squad	77.4	77.0
100%	ScholarlyRead	74.9	74.6

Table 4: **Comparison of Squad and ScholarlyRead dataset on constraint R_1 and R_2. Each score represents the average span F1 on our test set**

7.3 Results

We train the models using ScholarlyRead training data and monitor the training performance of the model on the validation set. We compare the results of the augmented models (i.e. BiDAF + R_1 and BiDAF + R_2) with the baseline model (i.e. BiDAF) on the ScholarlyRead test set. We report the final EM and F1 scores on the test set in Table 3. We can clearly see that incorporating logic in the neural network model improves the baseline performance by a significant extent. BiDAF with constraints R_1 and R_2 achieve F_1 scores as 74.6 and 74.9, respectively, which improves the F1 score almost by double over the BiDAF (baseline) model. The model shows the exact match (EM) scores of 60.8 and 61.0 for the constraints R_1 and R_2, respectively. This improves EM by +40.4 points over the BiDAF model. Rule R_1 is simple straight forward and another rule R_2 which is more conservative compared to R_1. Augmented model performs better with rule R_1 compared to the rule R_2. So the BiDAF with constraint R_1 achieves the best performance.

7.4 Comparing Systems

In this section, we compare the BiDAF and augmented BiDAF models. We describe each layer of the augmentation model in Section 1.5. The logic is augmented at the attention layer of the BiDAF model. The attention layer is guided using a knowledge-driven method. In the BiDAF model is computing the attention in two directions, i.e. *context-to-query* and *query-to-context*. We incorporate logic in the context to query attention weight and vice-versa.

7.5 Analysis

Table 2 presents the output of the BiDAF (Saikh et al., 2020) and logic augmented model on the ScholarlyRead dataset. It is observed that the logic augmented model resolves the type of errors that the BiDAF model encounters.

1. *External Knowledge*: Such errors occur when the model needs extra knowledge to understand the question and extract the answer from the paragraph. As shown in Table 2, example no. 4. In this example, the priority schema is not explained and not even mentioned. So to answer this question, we will need extra knowledge. In the augmented model, we solve such errors with the help of ConceptNet. Using the ConceptNet knowledge graph, we extract the relations between each word of the query and paragraph. Attention scores are calculated based on the similarity of the query and paragraph words. The best probability scores for the starting and ending indices of the answer in the paragraph are generated.

2. *Answer paraphrase problem*: Sometimes, the generated answer strings correspond to the paraphrase strings of the original ground truth. Table 2 shows example no. 3 of such problems. The correct answer is *network performance*, but the predicted answer by the BiDAF model is *network efficiency*. There are similarities in the meaning of the words *performance* and *efficiency*. This error is high due to the lack of knowledge of words related to the model. In the augmented model, we solve the paraphrasing problem with the help of the ConceptNet knowledge graph.

3. *Syntactic complications and ambiguities*: Sentence ambiguity is a challenge for the model while generating the answers. The attention-based model is unable to resolve semantic complexity correctly. In Table 2 example no. 2, the semantic complexity between the context word *network efficiency* and the query word *protocol*

is efficient. We have solved this type of error with the help of external knowledge (such as ConceptNet). The model needs extra knowledge for correct word alignment. We extracted the synset of each related word of context and query from external knowledge to learn the model better.

4. *Incorrect Pre-processing*: These types of errors occur when the dataset has noise or the tokens having special symbols (such as !, #, &, *) mismatch between the paragraph and answers tokens. In the given example of 'incorrect preprocessing' in Table 2 example no. 5, in BiDAF, the word *3G* is tokenized into two tokens (*3* and *G*). Due to this, there is a mismatch between the index of the answer on the given dataset and the index of the answer in the context file. Keeping this error in mind, we paid more attention to special symbols of context and answer files.

5. *Imprecise answer boundaries*: Only attention-based model cannot resolve this error. For this, we will need extra knowledge. Our Augmented Bidaf model can predict the boundary of the answer using external knowledge such as Conceptnet.

We also present the results of the augmented model for the *Squad* and *ScholarlyRead* datasets. The comparison of the results on both the dataset is shown in Table 4.

8 Conclusion and Future Work

In this paper, we present a logic-infused deep neural network for question answering on scholarly articles. We incorporate the logic at the attention layer of the popular BiDAF model. We also show how to convert first-order logic into the differentiable component without using extra learnable parameters. Our augmented models (BiDAF+R1 and BiDAF+R2) successfully outperform the baseline BiDAF model by a significant margin (+37.6 points improvement over the baseline model in terms of F1 score). We observe that the rule R_1 performs better with the BiDAF than the rule R_2. In the future, we will explore the effectiveness of infusing first-order logic into a dynamic language model like BERT for the same task. Research in this problem would prove helpful for the researchers who would want to survey multiple articles in a short time and thus accelerate the literature survey phase in the research life cycle.

References

Ali Mohamed Nabil Allam and Mohamed Hassan Haggag. 2012. The question answering systems: A survey. *International Journal of Research and Reviews in Information Sciences (IJRRIS)*, 2(3).

Martin Anthony. 2003. Boolean functions and artificial neural networks. *Boolean Functions*, 2.

Tarek R Besold, Artur d'Avila Garcez, Sebastian Bader, Howard Bowman, Pedro Domingos, Pascal Hitzler, Kai-Uwe Kühnberger, Luis C Lamb, Daniel Lowd, Priscila Machado Vieira Lima, et al. 2017. Neural-symbolic learning and reasoning: A survey and interpretation. *arXiv preprint arXiv:1711.03902*.

Ming-Wei Chang, Lev Ratinov, and Dan Roth. 2012. Structured learning with constrained conditional models. *Machine learning*, 88(3):399–431.

Nuri Cingillioglu and Alessandra Russo. 2018. Deeplogic: Towards end-to-end differentiable logical reasoning. *arXiv preprint arXiv:1805.07433*.

Andre Esteva, Anuprit Kale, Romain Paulus, Kazuma Hashimoto, Wenpeng Yin, Dragomir Radev, and Richard Socher. 2020. Co-search: Covid-19 information retrieval with semantic search, question answering, and abstractive summarization. *arXiv preprint arXiv:2006.09595*.

Deepak Gupta, Swati Suman, and Asif Ekbal. 2021. Hierarchical deep multi-modal network for medical visual question answering. *Expert Systems with Applications*, 164:113993.

Karl Moritz Hermann, Tomáš Kočiskỳ, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. *arXiv preprint arXiv:1506.03340*.

Zhiting Hu, Xuezhe Ma, Zhengzhong Liu, Eduard Hovy, and Eric Xing. 2016. Harnessing deep neural networks with logic rules. *arXiv preprint arXiv:1603.06318*.

Rudolf Kadlec, Martin Schmid, Ondrej Bajgar, and Jan Kleindienst. 2016. Text understanding with the attention sum reader network. *arXiv preprint arXiv:1603.01547*.

DP Kingman and J Ba. 2015. Adam: A method for stochastic optimization. conference paper. In *3rd International Conference for Learning Representations*.

Oleksandr Kolomiyets and Marie-Francine Moens. 2011. A survey on question answering technology from an information retrieval perspective. *Information Sciences*, 181(24):5412–5434.

Tao Li and Vivek Srikumar. 2019. Augmenting neural networks with first-order logic. *arXiv preprint arXiv:1906.06298*.

Maria Liakata, Simon Dobnik, Shyamasree Saha, Colin Batchelor, and Dietrich Rebholz Schuhmann. 2013. A discourse-driven content model for summarising scientific articles evaluated in a complex question answering task. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 747–757.

Hugo Liu and Push Singh. 2004. Conceptnet—a practical commonsense reasoning tool-kit. *BT technology journal*, 22(4):211–226.

Kaixin Ma, Jonathan Francis, Quanyang Lu, Eric Nyberg, and Alessandro Oltramari. 2019. Towards generalizable neuro-symbolic systems for commonsense question answering. *arXiv preprint arXiv:1910.14087*.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Tanik Saikh, Asif Ekbal, and Pushpak Bhattacharyya. 2020. Scholarlyread: A new dataset for scientific article reading comprehension. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 5498–5504.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2016. Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603*.

Shashank Srikant and Una-May O'Reilly. 2021. Can cognitive neuroscience inform neuro-symbolic inference models? In *Is Neuro-Symbolic SOTA still a myth for Natural Language Inference? The first workshop*.

Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. 2015. Highway networks. *arXiv preprint arXiv:1505.00387*.

Ricardo Usbeck, Michael Röder, Michael Hoffmann, Felix Conrads, Jonathan Huthmann, Axel-Cyrille Ngonga-Ngomo, Christian Demmler, and Christina Unger. 2019. Benchmarking question answering systems. *Semantic Web*, 10(2):293–304.