Textual Question Answering for Semantic Parsing in Natural Language Processing

Conference Paper · September 2019

DOI: 10.1109/ICASERT.2019.8934734

3 authors:

Jaydeb Sarker
Wayne State University

Mustain Billah
Jessore University of Science and Technology

Md. Al Mamun
Rajshahi University of Engineering & Technology

Textual Question Answering for Semantic Parsing in Natural Language Processing

Jaydeb Sarker
Dept. of CSE
Rajshahi University of Engineering and Technology
Rajshahi, Bangladesh
jaydeb.ruet@gmail.com

Mustain Billah
Dept. of IT
University Of Information Technology and Sciences
Dhaka, Bangladesh
mustain.billah@uits.edu.bd

Md. Al Mamun
Dept. of CSE
Rajshahi University of Engineering and Technology
Rajshahi, Bangladesh
a.mamun@cse.ruet.ac.bd

***Abstract*—Question answering (QA) problem is one of the most frequent tasks in natural language processing (NLP) over language input. To understand the natural language, semantic parsing is an extensively used process. For deriving syntactic and semantic structures of texts, synchronous frameworks are being used frequently. With the aim of answering questions efficiently, in this paper, an approach by semantic parsing of texts with a view to leveraging semantic information is introduced. The proposed approach uses the method based on lambda calculus for semantic parsing to derive logical forms of sentences. The questions are analyzed by collecting significant features to find out correct answers from the facts. Unlike traditional approaches, this paper extends lambda calculus based dependency parser methodology for question analysis. This proposed approach for answering questions is based on generalized searching techniques. Proposed system achieves 95% accuracy for Yes/No questions with an overall 83% mean accuracy for the five tasks of bAbI-10k question answering dataset surpassing the existing approach by about 11%.**

***Index Terms*—Natural Language Processing, Semantic Parsing, Lambda Calculus, Question Answering, Logical Forms, Text Parsing.**

I. Introduction

A complicated task in natural language processing [1]–[4] is question answering which requires the understanding of the meaning of a text and the ability to reason over relevant facts. To represent the meaning of sentences of the human language, semantic parsing is the best-defined tasks that support automated inference for knowledge representation [5], [6]. Relevant works includes Transforming Dependency Structures where they use lambda calculus for parsing method to find dependency of sentences [7]. The dependency of words in a sentence can be found by open text semantic parsing [6].

For building a question answering system, it is important to analyze the question and input text; then the answer has been found from the relevant facts. One of the popular question answering task is factoid question answering to find the correct answer in documents [7]. Dynamic memory network, a neural network architecture has been introduced by Kumar et al. that processes input sequences and questions, forming episodic memories, and generates relevant answer [8]. Query-Regression Network (QRN) approach, a variant of recurrent neural network (RNN), is able to effectively handle long-term dependencies and is highly parallelizable [9]. This model is also suitable for end-to-end machine comprehension. Todai robot project on Japanese examination system has been learned the exam answers and test that from the student answers [10]. Question answering by using syntactic information is also enhanced in several attempts using heuristics analysis for ranking potential answers [11]. Some interesting works on question answering system are found in [14]–[16], [20]. QA module has been also designed for business module [15].

In the question answering system, syntactic information gives good accuracy than information retrieval. In proposed approach both sentences and selected questions in the top documents are parsed and stored in their corresponding database. We propose a novel technique to find out the answer to questions by a generalized searching algorithm. Our method first analyzes the input text and questions and find the relevant facts. It finds the answer by searching technique from the relevant facts. Fig.1 provides an example for question answering technique:

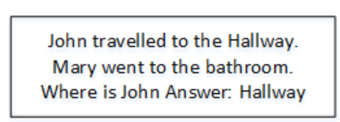

Fig. 1. Example of inputs and questions, together with answers.

This paper consists of following sections: Proposed structure and methods are discussed in section II. Section III contains the results and discussions. Finally, the paper is concluded in section IV.

II. Proposed Structure and Methods

Proposed system introduces a searching based novel approach to find the answers from corresponding text. However, it consists of four modules: Input module, Facts finding module, Question Module and Answer module (figure 2). Proposed system is designed to map the sentences to its logical forms. Features from questions are found and the actual answers are extracted by searching from facts.

It first computes a representation for all inputs and the questions. The question representation has been triggered an iterative attention process that searches the inputs and retrieves relevant facts.

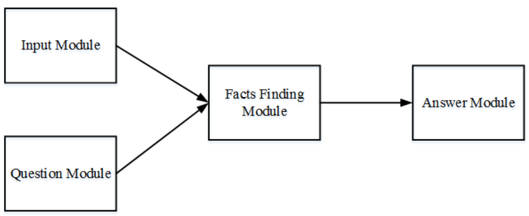

Fig. 2. Proposed Question Answering System

A. Input module

In the input module, a set of text is given. The text represents a long sequence of sentences of a story or movie. All the sentences are in simple sentence structure which have sequential meaning. Firstly, each of the sentence is being parsed from the text by nltk POS tagger [13]. Logical forms of each sentence are derived from lambda calculus based dependency parsing method [12]. Lambda calculus based dependency parsing method has been chosen for parsing because it puts the main $verb$ as root word of a sentence. The semantic form of each sentence depends upon the $verb$. The $subject$ and $object$ of a sentence have been affiliated by verb.

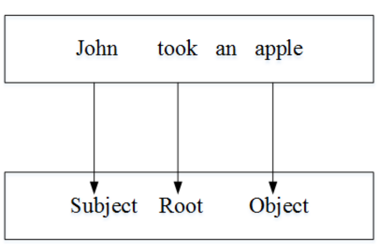

Fig. 3. Lambda calculus parsing method.

The facts are found from the text by parsing method based on lambda calculus dependency parsing [12]. Lambda calculus method has been used for making a relationship with $subject$ and $object$ of a sentence by taking the verb as a root of the sentence. An example of lambda calculus method is depicted in fig. 3.

B. Facts finding module

The given input text has been analyzed in this module. The $subject$ and $object$ of the sentence are also derived by checking tagged parts of speech of each words of that sentence. The $verb$, $subject$ and $object$ from each sentence have been taken and stored in corresponding database for easily finding the answer.

Logical forms derived from lambda calculus dependency parsing [12] will help further to find the features those eventually lead us to analyze question to find answers.

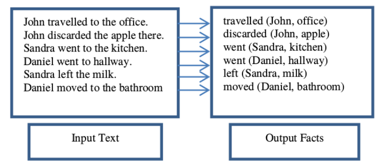

Fig. 4. Lambda calculus parsing method is used to find facts from input text.

In fig. 4, a sequence of input sentences have been parsed by lambda calculus dependency parsing method. Three types of feature are found from each sentence. The extracted features are stored in their corresponding databases, like verb database (VB_i), subject database (SB_i) and object database (OB_i). It assumes that if there are n number of sentences in a text, then the number of $verb$, $subject$ and $object$ are also n numbers. Here *i* represents the position of sentences in each text. Answer of the question has been extracted from these databases. The database of $verb$, $subject$ and $object$ extracted from fig. 4 has been shown in fig. 5, 6, 7.

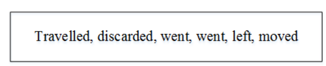

Fig. 5. Verb database (VB_i) for corresponding input text.

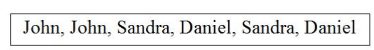

Fig. 6. Subject database (SB_i) for corresponding input text.

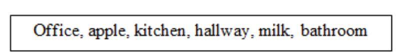

Fig. 7. Object database (OB_i) for corresponding input text.

C. Question Module

The corresponding question is also expressed in natural languages which is related to the input text. We mainly work on analyzing two types of question. They are 'W/H' question and 'Yes/No' question. 'W/H' questions are started with 'WH' tagged word by using the natural language toolkit (NLTK) [13]. Example:

$$Where\ is\ Daniel?$$

The answer of the question which starts with $Where$, indicates that the answer should be a place. Similarly, ‘W/H’ question extracts the factual answers from the text. On the other hand, ‘Yes/No’ question starts with a auxiliary verb (like, am, is, are etc). This type of question only requires ‘Yes’ or ‘No’ as answers.

After parsing and tagging each word of a question, we find two features from that sentence.

- Question Word (QW)
- Object Word (OW)

Question represents a sentence according to the input text. For example $Where\ is\ Milk?$ is an input question. Question should be corresponding to the input sequence. The important features from the question have been derived by extracting the features which is depicted in fig. 8. The two important features (w/h question word and object word) have been derived from each question.

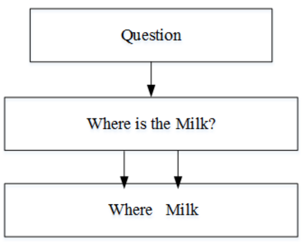

Fig. 8. Feature finding from W/H question.

Let we take two main features in QW(question word) and OW(object word). Fig. 8 shows that
QW = ‘where’
OW = ‘milk’
QW is also known as W/H feature. W/H feature is ‘where’ which indicates to find the location of the milk.

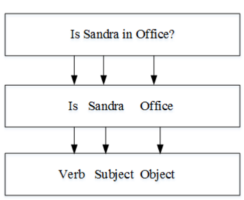

Fig. 9. Feature finding from Yes/No question.

Similarly, Figure 9 shows the feature finding from ‘Yes/No’ question word. Two features ($Subject$ and $Object$) are collected from this ‘Yes/No’ type question.

D. Answer module

This module generates the answer analyzing by the facts those are found from input module and question module. We have found two types of question in our test from the dataset. Firstly, we made a database of verb which indicates the position of that subject. For example, $Daniel\ Moved\ to\ office$. In this case, $moved$ indicates the location of the subject of this sentence. The positional verb database T_i is shown in figure 10.

Moved, went, stayed, Travelled, Journeyed

Fig. 10. Positional verb database T_i.

Storage of positional verb helps us to find the answer of a question which starts with ‘where’. For finding the last position of the given Object Word (OW) in the question, it is important to search from the last position of the database and find the matches. The question which is started with ‘where’, our proposed system finds the answer by the given algorithm:

Algorithm 1: Finding answers of Questions starting with “Where”

Input: OW
Output: Position of SBi (where i is the position of sentence)

```
Take OW from the question
Check OW with SBi and VBi with Ti,
  answer = OBi
if OW matches with OBi then
    position of SBi of corresponding OBi is found by
      previous step.;
end
```

To test our algorithm, we give two examples here. For instance, the question is $Where\ is\ Sandra?$. Here, the OW = ‘Sandra’ and by using our algorithm the answer is ‘kitchen’. It generates ‘kitchen’ as answer because the last position of Sandra is in ‘kitchen’.

We also design an algorithm for ‘Yes/No’ question answering. It is known to us that two features ($Subject$ and $Object$) are collected from the corresponding question for ‘Yes/No’ answers. The algorithm is as following:

Algorithm 2: Finding answers of Yes/No Questions

Input: Subject, Object
Output: Answer

```
Take Subject and Object from the question
if Subject = SBi and Object = OBi (where i is the
  position of sentence) then
    answer = 'Yes';
else
    answer = 'No'
end
```

For example, the question is $Is\ Sandra\ in\ Office$, where $Subject$ = ‘Sandra’ and $Object$ = ‘Office’. By using our proposed algorithm, it finds no matches where ‘Sandra’ and ‘Office’ occurs in the same sentence. So, the system gives ‘No’ as it's answer. Moreover, we also make the similar matching algorithm for finding the answer of question which starts with $what$.

III. Results And Discussion

A. Dataset Collection

To analyze proposed model, bAbI-10k English question answering dataset [9] has been used. It's a synthetic dataset featuring 20 different tasks. The dataset comes in two sizes: bAbI-1k and bAbI-10k, referring to the number of training examples of each task. Proposed system is tested by 5 different tasks of bAbI-10k. Five tasks are listed with their corresponding class name in table I.

TABLE I
THE TASK IN BABI/TASKS CORRESPOND TO THOSE FROM THE ORIGINAL DATASET AS FOLLOWS:

	Task	Class Name
1	QA with single supporting fact	WhereIsActor
2	QA with two supporting fact	WhereIsObject
3	QA with three supporting fact	WhereWasObject
4	Yes/No questions	IsActorThere
5	List/Sets	Listing

In case of single supporting facts, only the questions about the position of the $Subject$ are asked. Tasks with two supporting facts have class name $Where\ Is\ Object$ which indicates to find out the location of the object. But, three supporting facts uses the sequential sentences and the question is about the position of object using the word either $before$ or $after$. On the other hand, Yes/No questions just justify the truth value of the question. Another task is about Lists/Sets, where more than one answer would be generated from the question.

B. Experimental Analysis

Analyzing each questions using proposed algorithm 1 and algorithm 2, answers are found for each question. For Parts of Speech Tagging, proposed system uses the NLT POS tagger [13]. As there is no need to update memory like dynamic memory network [8] or QRN [9] , it is easy to find the relevant facts without any complexity. It can retrieve answers easily by searching from the databases of facts.

C. Performance Evaluation

Accuracy of proposed Question Answering system for different tasks are shown in TABLE II. Better accuracy have been gained for each tasks excepts Lists/Sets. TABLE II also shows the number of input sentences and questions for testing the system. The proposed system gives the highest accuracy (95%) for Yes/No questions. Without the Lists/Sets task, the system gives more than 80% correct answer for given questions. The performance degrades when new type of tasks are included as input in the system. Though it gives high performance for sequence of simple sentences but for compound and complex sentences, it will be tough to find the features by lambda calculus based dependency parsing method.

TABLE II
ACCURACY OF QA FOR DIFFERENT TASKS

	Task	No of Sentences	No Questions	System Accuracy
1	Single Supporting Facts	13	10	90%
2	Two Supporting Facts	25	10	90%
3	Three Supporting Facts	37	10	80%
4	Yes/No questions	40	20	95%
5	List/Sets	60	10	60%

Table III and Figure 11 show the comparison of Structured Support Vector Machine(SSVM) [14] and proposed system. Our proposed QA system gives a better accuracy than SSVM for Two and Three supporting facts. To be precise, our system gives 80% accuracy for Three Supporting facts whereas SSVM gives only 17%. The overall mean accuracy of our system is 83% for these five types of tasks which surpasses the SSVM. The mean accuracy of SSVM is 71.8% for this five types of tasks.

TABLE III
COMPARISON OF STRUCTURED SVM AND PROPOSED SYSTEM

	Task	Structured SVM	Proposed System
1	Single Supporting Facts	99%	90%
2	Two Supporting Facts	74%	**90%**
3	Three Supporting Facts	17%	**80%**
4	Yes/No questions	99%	95%
5	List/Sets	70%	60%
	Mean Accuracy	71.8%	**83%**

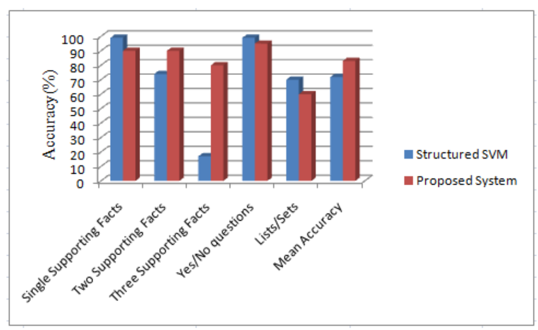

Fig. 11. Accuracy of Structured SVM vs Proposed System

We have considered 5 specific tasks for our work. For these tasks, our system shows better accuracy than some other existing methods. Most of the existing methods used machine learning approach which requires a large data sets. On the other hand, proposed method performs well for small dataset which resembles real life problems. Performance comparison with existing methods [14] is shown in Table IV.

IV. CONCLUSION AND FUTURE WORK

In this paper, a new methodology of question answering system has been introduced. POS tagging and semantic parsing techniques have been used for finding the answers from a given text. Proposed system is easy and less complex than existing systems in case of simple sentences. Moreover, it gains a better accuracy for five types of tasks in question answering

TABLE IV
PERFORMANCE COMPARISION WITH EXISTING METHODS

Methods	Accuracy
N-gram	26.8%
LSTM	36.6%
MemNN	68.8%
Proposed Method	**83%**

systems than the existing system. However, proposed question answering system is implemented only for simple sentences. In future, we will work with compound and complex sentences and long sequence of text. Machine learning technique with position vector will be used to learn the input text and retrieve the answer.

ACKNOWLEDGEMENT

We are very grateful to Biprodip Pal, Assistant Professor, Dept of CSE, Rajshahi University of Engineering Technology (RUET), Md Al Shayokh, Assistant Professor, Dept of IT, University of Information Technology and Sciences (UITS) and Syntax and Semantic Research Group (SySeReG) for their kind cooperation and contribution.

REFERENCES

[1] Fonferko-Shadrach, Beata, et al. “Using natural language processing to extract structured epilepsy data from unstructured clinic letters.” International Journal of Population Data Science 3.4 (2018).
[2] Weston, Jason E., et al. “Techniques to predictively respond to user requests using natural language processing.” U.S. Patent Application No. 15/077,814.
[3] Nie, Pengyu, et al. “Natural language processing and program analysis for supporting todo comments as software evolves.” Workshops at the Thirty-Second AAAI Conference on Artificial Intelligence. 2018.
[4] Weston, Jason E., et al. “Techniques to predictively respond to user requests using natural language processing.” U.S. Patent Application No. 15/077,814.
[5] Dreisbach, Caitlin, et al. “A systematic review of natural language processing and text mining of symptoms from electronic patient-authored text data.” International Journal of Medical Informatics (2019).
[6] Kaur, Parneet, Vandana Pushe, and Rupinderdeep Kaur. “A Review of NLP Based Upon Semantic Analysis.” (2014).
[7] Wang, M., “A survey of answer extraction techniques in factoid question answering,” Computational Linguistics 1.1, vol. 9, 2006.
[8] Xiong, C., S. Merity, and R. Socher, “Dynamic memory networks for visual and textual question answering ,” arXiv preprint arXiv:1603.01417, vol. 534, 2016.
[9] Seo, M., H. Hajishirzi, and A. Farhadi, “Query-Regression Networks for Machine Comprehension,” arXiv preprint arXiv:1606.04582, vol. 534, 2016.
[10] Fujita, Akira, et al. “Overview of todai robot project and evaluation framework of its nlp-based problem solving.” World History 36 (2014): 36.
[11] Li, X.,“Syntactic features in question answering, ” Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval. ACM, vol. 38, 2003.
[12] Reddy, Siva, O. Tckstrm, M. Collins, T. Kwiatkowski, D. Das, M. Steedman, and M. Lapata. “Transforming Dependency Structures to Logical Forms for Semantic Parsing,” Vol.: 38, 2016.
[13] Loper, E., and S. Bird, “NLTK:The natural language toolkit.” Proceedings of the ACL-02 Workshop on Effective tools and methodologies for teaching natural language processing and computational linguistics-Volume 1. Association for Computational Linguistics, 2002.
[14] Weston, Jason, et al. “Towards ai-complete question answering: A set of prerequisite toy tasks.” arXiv preprint arXiv:1502.05698 (2015).
[15] Curtis, Jon, Gavin Matthews, and David Baxter. “On the effective use of Cyc in a question answering system.” Proc Workshop on Knowledge and Reasoning for Answering Questions. 2005.
[16] Sukhbaatar, Sainbayar, Jason Weston, and Rob Fergus. “End-to-end memory networks.” Advances in neural information processing systems. 2015.
[17] Chung, Junyoung, et al. “Empirical evaluation of gated recurrent neural networks on sequence modeling.” arXiv preprint arXiv:1412.3555 (2014).
[18] Graves, Alex. “Generating sequences with recurrent neural networks.” arXiv preprint arXiv:1308.0850 (2013).
[19] Rajpurkar, Pranav, et al. “Squad: 100,000+ questions for machine comprehension of text.” arXiv preprint arXiv:1606.05250 (2016).
[20] Rahman, Habibur, Julia Rahman, and Asmaul Husna. ”WALGES: Weighted Probability Based Scoring Approach for Solving Algebraic Word Problems using Semantic Parsing.” 2019 International Conference on Electrical, Computer and Communication Engineering (ECCE). IEEE, 2019.