Automatic Question and Answer Pair Generation

K.Surya, R.Aarthipriya , S.Iswarya
Students, Department of Computer Science and Engineering, Periyar Maniammai University, Thanjavur, India

Abstract: Our new system generates the question and answer pair automatically. Using open NLP Tool, we can create a new system. The computer system first analyses the given input sample or document, then it produces the valuable questions and also their appropriate answers using patterns, machine learning algorithm and natural language processing. Our goal is to understand what level of the language understanding is required to perform this task. Our system first analyses the document and then creates various question about document together with answer to those questions. Our system reduces the dependency upon human being and also very useful for institution to avoid repetition of question and answer pair.

Keywords: *Machine learning, Natural language processing, Pattern, Reading comprehension, Question answering*

I. INTRODUCTION

Machine learning focuses on the development of computer programs that can teach themselves to grow and change when exposed to new data. Our project generates the question and answer pairs automatically. It is very useful for school going students. Machine learning is a type of artificial intelligence (AI) that provides computers with the ability to learn without being explicitly programmed. Natural language processing is the interactions between computer and human languages. It is the ability of a computer program to understand text and speech .It deals with analysing, understanding and generating the languages. Reading comprehension (RC) is the ability to read text, process it, and understand its meaning. Reading comprehension is often tested by asking questions that require interpretive understanding of a passage. In recent years, there have been several strands of work which attempt to collect human-labeled data for this task , in the form of document, question and answer triples and to learn machine learning models directly. However, these datasets consist of only hundreds of documents, as the labeled examples usually require considerable expertise and neat design, making the annotation process quite expensive. The subsequent scarcity of labeled examples prevents us from training powerful statistical models, such as deep learning models, and would seem to prevent a system from learning complex textual reasoning capacities. We describe two systems 1.conventional entity-centric classifier and 2.an end-to-end neural network to provide several baselines for performance on the RC task, we suspect that their baselines are not that strong. They attempt to use a frame-semantic parser, and we feel that the poor coverage of that parser undermines the results, and is not representative of what a straightforward NLP system based on standard approaches to factoid question answering and relation extraction developed over the last 15 years can achieve. Indeed, their frame-semantic model is markedly inferior to another baseline they provide, a heuristic word distance model. At present just two papers are available presenting results on this RC task, both presenting neural network approaches While the latter is wrapped in the language of end-to-end memory networks, it actually presents a fairly simple window-based neural network classifier running on the CNN data. Its success again raises questions about the true nature and complexity of the RC task provided by this dataset, which we seek to clarify by building a simple attention-based neural net classifier. Our fully implemented system first analyzes the structure of the input text and then creates various question-answer pairs using patterns. It Pre-select and introduce the text to be used for generating questions.

II. RELATED WORKS

At university of Pennsylvania [1] the system designed to our work. Using rhetorical analysis, it generates questions from text. For analysing the meaning of the text, they use semantic role labelling. For analyzing the discourse structure of the text [2], using support vector machine classifiers. When considering question generation at paragraph level the discourse structure of the text becomes important [5]. The main aim of the children book test[3] to improve reading comprehension of children. Instead of discourse analysis and dialogue generation, their approach is applying semantic role labelling [1] for generating questions. Further, their generated questions are used as a tool for classification of children self-questioning responses, whereas our generated question-answer pairs are used as input for text book. Cloze question generation is based on syntactical analysis [4], and takes a similar approach as our work. Trees are constructed, patterns are matched and questions are generated.

Richardson et al., 2013 did the MCTest [6] which is open domain reading comprehension task in the form of fictional short stories accompanied by multiple choice questions. It was created by using crowd sourcing, and aims at a 7-year-old reading comprehension level. On the one hand, this dataset has a high demand on various reasoning capacities: over 50% of the questions require multiple sentences to answer and also the questions come in assorted categories (what, why, how, whose, which, etc). On the other hand, the full dataset has only 660 paragraphs in total (each paragraph is associated with 4 questions), which renders training statistical models (especially complex ones) very difficult.

Test Hill et al., 2016 was developed in a similar spirit to the CNN/Daily Mail datasets[7] in Children Book. It takes any consecutive 21 sentences from a children's book – the first 20 sentences are used as the passage, and the goal is to infer a missing word in the 21st sentence (question and answer). The questions are also categorized by the type of the missing word: named entity, common noun, preposition or verb. According to the first study on this dataset [7], a language model (an n-gram model or a recurrent neural network) with local context is sufficient for predicting verbs or prepositions; however, for named entities or common nouns, it improves performance to scan through the whole paragraph to make predictions. So far, the best published results are reported by window-based memory networks.

III. PROPOSED METHOD

Our work is fully based on processing of natural language text. It includes a sentence detector, a tokenizer, a name finder, a parts-of-speech (POS) tagger, Question and answer pair generator using NLP Tool. Figure 1 shows the block diagram of our project. Our goal is to Enabling a computer to understand a document so that it can answer comprehension questions. A key factor impeding its solution is the limited availability of human-annotated data by machine learned systems.

We employ the following algorithm for this system

1. Split the document content into sentence.
2. Split the sentence into several words
3. Identify their names in their context
4. Identify the words based on particular part of speech
5. Generate question and answer pairs

A. Sentence Detector

Sentence detector is for detecting sentence boundaries from the given input text. It returns an array of string.

The given input sample:

Brian is a doctor. He looks after sick people. He usually gets up at 6.00 o'clock. Today he is late, it is 6.30 and he is still in bed. He usually goes to work by train but today he is driving to work. He arrives at work at 6.30 every morning,but he is still driving 7.30

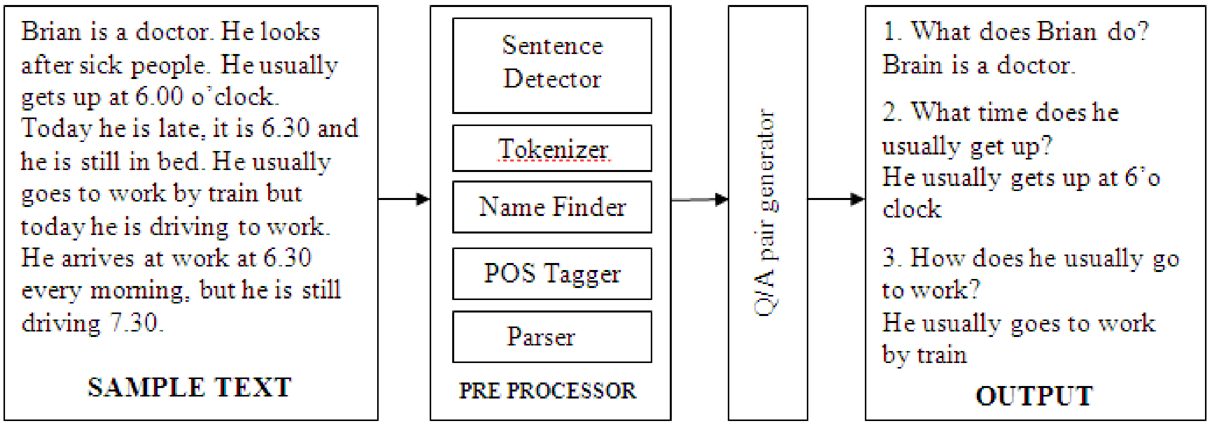
Brian is a doctor. He looks after sick people. He usually gets up at 6.00 o'clock. Today he is late, it is 6.30 and he is still in bed. He usually goes to work by train but today he is driving to work. He arrives at work at 6.30 every morning, but he is still driving 7.30.
SAMPLE TEXT
Sentence Detector
Tokenizer
Name Finder
POS Tagger
Parser
PRE PROCESSOR
Q/A pair generator
1. What does Brian do?
Brain is a doctor.
2. What time does he usually get up?
He usually gets up at 6'o clock
3. How does he usually go to work?
He usually goes to work by train
OUTPUT

Figure 1: Block Diagram For Q/A Pair Generation

After Sentence Detector process method:

Brian is a doctor.

He looks after sick people.

He usually gets up at 6.00 o'clock.

Today he is late, it is 6.30 and he is still in bed.

He usually goes to work by train but today he is driving to work.

He arrives at work at 6.30 every morning, but he is still driving 7.30

B. Tokenizer

Tokens are usually words which are separated by space, but there are exceptions. For example, "isn't" gets split into "is" and "n't, since it is a a brief format of "is not". In this Process, the above sentence is separated into the tokens.

C. Name Finder

By its name, name finder just finds names in the context. Check out the following example to see what name finder can do. It accepts an array of strings, and find the names inside.

D. POS Tagger

It is the process of marking up a word in a text as corresponding to a particular part of speech based on both its definition and its context. It identify the words such as nouns, verbs, adjectives, adverbs etc This process identify the part of speech in the given input sentence.

Brian_NNP is_VBZ a_DT doctor_NN. He_PRP looks_VBZ after_IN sick_JJ people_NNS . He_PRP usually_RB gets_VBZ up_RP at_IN 6.00_CD o'clock_NN ._. Today_NN he_PRP is_VBZ late_RB ,_, it_PRP is_VBZ 6.30_CD and_CC he_PRP is_VBZ still_RB in_IN bed_NN ._. He_PRP usually_RB goes_VBZ to_TO work_VB by_IN train_NN but_CC today_NN he_PRP is_VBZ driving_VBG to_TO work_VB ._. He_PRP arrives_VBZ at_IN work_NN at_IN 6.30_CD every_DT morning_NN ,_, but_CC he_PRP is_VBZ still_RB driving_VBG 7.30_CD

E. Chunker

Chunker may not be a concern for some users, but it is worth to mention it here. What chunker does is to partition a sentence to a set of chunks by using the tokens generated by tokenizer.

F. Parser

Given this sentence: "Programcreek is a very huge and useful website.", parser can return the following:

```
|(TOP
    (S
        (NP
        (NN ProgramCreek)
    )
    (VP
        (VBZ is)
        (NP
            (DT a)
            (ADJP
                (RB very)
                (JJ Huge)
                (CC and)
                (JJ Useful)
            )
        )
    )
    (.website.)
    )
)
```

G. Q/A Pair Generator

This Module is used to generate the Q/A Pair based on the output of parser.

CONCLUSION

We examined the reading comprehension task and proposed a new technique to generate the question and answer pairs based on the input datasets which creates number of valuable questions for the given text and also gives the proper answers. The proposed system can be used for any educational institutions which makes easy the work of preparing question papers and keys. In future, we will utilise the datasets which includes image data.

References

[1] Mannem, P., Prasad, R.P., Joshi, *A. Question Generation from Paragraphs at UPenn. In: Proc. of QG2010: The Third Workshop on Question Generation*, pp. 84-91, Pitsburg, PA (2010)

[2] Hernault, H., Prendinger, H., duVerle, D. and Ishizuka, M. HILDA: *A Discourse Parser Using Support Vector Machine Classification. In: Dialogue and Discourse*, vol. 1, no. 3, pp. 1-33 (2010)

[3] Chen, W., Mostow, J. Aist, G. *Using Automatic Question Generation to Evaluate Ques-tions Generated by Children. In: Question Generation: Papers* from the 2011 AAAI Fall Symposium (2011)

[4] Gates, D., Aist, G., Mostow. J., McKeown. M., Bey, J. *How to Generate Cloze Questions from Definitions: A Syntactic Approach. In: Question Generation:* Papers from the 2011 AAAI Fall Symposium (2011)

[5] Heilman, M. *Automatic Factual Question Generation from Text.* In: Ph.D. Dissertation, Carnegie Mellon University, CMU-LTI-11-004CMU-LTI-09-013 (2011)

[6] Matthew Richardson, Christopher J.C. Burges, and Erin Renshaw. 2013. MCTest: A challenge dataset for the open-domain machine comprehension of text. In Empirical Methods in Natural Language Processing (EMNLP), pages 193–203.

[7] Felix Hill, Antoine Bordes, Sumit Chopra, and Jason Weston. 2016. The goldilocks principle: Reading children's books with explicit memory representations. In nternational Conference on Learning Representations (ICLR)