Original Research Article

J. Computing & Int. Systems (2024) 351–355

SACRED HEART RESEARCH PUBLICATIONS

Journal of Computing and Intelligent Systems

Journal homepage: www.shcpub.edu.in

ISSN: 2456-9496

AUTOMATIC QUESTION GENERATION SYSTEM USING MACHINE LEARNING TECHNIQUES

Poovizhi Magendiran #1

Received on 05 MAR 2024, Accepted on 20 MAR 2024

Abstract — The automatic question generation is to generate the possible question from the given document. It reads the document and identifies the target words from the sentence to generate questions using Natural Language Processing (NLP) which is a domain that makes the machine understand the human language. This system could also generate taxonomy cognitive level questions. Various methods are employed to formulate the query based on the provided passage. The automatic question generation based on the Part-of-Speech (POS) probability of the previously trained data is presented in this article.

Keywords - Automatic Question Generation, Bloom's Taxonomy, Recurrent Neural Network (RNN)

I INTRODUCTION

The process of automatic question generation contains many tasks. Selecting the targeted sentence and targeted words to generate the question. Before the targeted words could be selected it should be switched into a machine-understandable structure using some text preprocessing techniques in Natural Language Processing (NLP).

The question will be generated using a pre-trained model using deep learning and machine learning techniques and the sentences will be formed based on the probability of the part-of-speech and the prediction of the next word in the sentence Using RNN (Recurrent Neural Network). The generated questions may be in unstructured grammar. Natural Language Tool Kit (NLTK) is a widely used package to work with Natural Language Processing (NLP).

II RELATED WORKS

Robert M.Losse explained the idea about decision making based on part-of-speech using retrieval information phrases to contribute grammatical constructor form low to high [1].

Human peer and generic questions were suggested by Ming Liu et al. in the majority of quality metrics after grammatical and semantic errors were removed from the questions. They also talked about how the human supervisors create these questions from the source text and determined the most common question types that they derived from the questions of the human supervisors. [3].

Du at el proposed the end-to-end and sequence-to-sequence learning to generate a question for reading comprehension and to generate the question-answer pair from the unstructured data using a rule-based approach that could generate WH-Questions [4].

Vishwajeet Kumar et al introduced the cross-lingual QG model. They used the unsupervised pertaining of the language model to generate questions in Hindi. QG models can be built from the existing question answering dataset [6].

* Corresponding author: E-mail: [1]poomca12@gmail.com

[1] Department of Computer Science, Sacred Heart College, Thiruvalluvar University, Tirupattur, Tamil Nadu, India.

Bloom's classification of learning outcome statements was enforced by Brain von Konsky et al., who also enforced automatic identification of verbs and other speech components that impact linguistics. Machine learning techniques were combined with table operations to automatically classify Bloom's levels [7].

Qingyu Zhou et al brought the idea about generating questions from the text with no predefined rules. It focuses on only the targeted input sentence to generate questions using the lexical features including part-of-speech and named entity tags [8].

III AQGS WORKFLOW

AQG system reads the document and extracts the keywords from the document to generate questions the models will be loaded from the local disk which helps us to get the results in a short time. The randomly selected keyword from the predefined keyword list of Bloom's Taxonomy that is proposed by Benjamin Bloom and the next word will be predicted using the selected keyword and it will be structured using the pre-trained model of the POS tag identifier.

Figure 1 represents the AQG workflow.

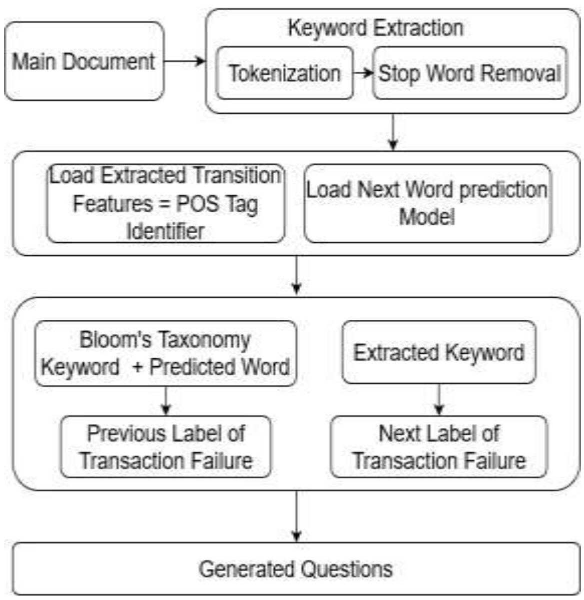

Figure – 1: AQG Workflow

IV EXTRACT KEYWORD

Keyword extraction is an important process in the automatic question generation system. This can be done with much Deep Learning and Machine Learning Technique i) Simple Statistical Approach ii) Linguistic Approach iii) Graph-based Approaches iv) Machine Learning Approach v) Hybrid Approaches these are the most used techniques to extract the keywords form the document. But here in our system, the keywords can be identified by the font size and the font style of the text in the document it is fit for the structured document. If the document unstructured one of the above text can be used to extract the keywords. The question which will be generated classified based on the weight of the paragraph and the threshold value.

If the Head Title of the document contains the paragraphs with contains quite fivehundred words thatexactly classifie d beneath ten or eight mark queries. Likewise, other questions also will be categorized under the threshold values. The sentence which got the highest font size will be the base key to identify other question keywords.

The Generative Question will make a sentence and maintain the structure of it by pre-trained model of POS (part-of-speech) tagging from the NLTK (Natural Language Tool Kit) corpus dataset to predict the part-of-speech of next word. It identifies the patterns from the trained dataset and predicts the POS tags to generate the sentence which does not follow any grammar rules.

V PART-OF-SPEECH (POS) IDENTIFIER

The Conditional Random Fields (CRF) is a Discriminative Probabilistic Classifier used to construct the model and to identify the next word of the part-of-speech (POS) from the trained data. Which is working using a conditional probability distribution. In the ANQG system, 1000 tagged sentences are used to build the model to get the occurrence of high-frequency POS tag using the previous word in the sentence

Word features in a sentence can be extracted using Conditional Random Fields (CRF). Table 1 displays the feature functions, also referred to as state features.

Table – 1: State Features

S.No	Features
1	is_first_capital
2	is_first_word
3	is_last_word
4	is_complete_capital
5	prev_word
6	next_word
7	is_numeric
8	is_alphanumeric
9	prefix_1
10	prefix_2
11	prefix_3
12	prefix_4
13	suffix_1
14	suffix_2
15	suffix_3
16	suffix_4
17	suffix_5
18	word_has_hyphen

Table – 2: Transition Features

S. No	Previous Label	Actual Label
1	TO	VB
2	MD	VB
3	NN-TL	IN-TL
4	JJ	NN
5	HL	NP-HL
6	QL	RB
7	QL	JJ
8	JJ	NNS
9	NP-HL	-HL
10	JJ-TL	NP-TL

When using CRF, the weights of the previous label word are transferred to the current word. The training data's likelihood of the labels will be maximized by CRF's attempt to control the weights of various feature functions. Transition Feature is the feature function that is reliant on the label of the preceding word. From the trained model, Table 2 displays the top 10 frequently occurring Transition Features.

The following components are used to train the model to identify the POS tags.

- **Number of Tagged Sentences: 1000**
- **Total Number of Tagged words: 22079**
- **Vocabulary of the Corpus: 4641**
- **Number of Tags in the Corpus: 147**
- **Number of Sentences in Training Data: 800**
- **Number of Sentences in Testing Data: 200**

To assess the model's performance using the F1 score, Precision, and Recall metrics. A few of the model's assessed examples are displayed in Table 3.

Table – 3: Precision, Recall, and F1-Score of the model.

S.No	Labels	Precision	Recall	F1-Score	Support
1	AP-HL	0.000	0.000	0.000	1
2	IN-HL	0.000	0.000	0.000	2
3	VBG-HL	0.000	0.000	0.000	2
4	NNS	0.888	0.990	0.936	208
5	AT	0.993	1.000	0.996	414
6	NN$	1.000	1.000	1.000	9
7	JJ	0.848	0.816	0.832	185
8	NN	0.902	0.952	0.926	607
9	IN	0.957	0.962	0.959	468
10	VBZ	1.000	0.583	0.737	24
11	micro avg	0.926	0.928	0.927	
12	macro avg	0.547	0.505	0.515	
13	weighted avg	0.918	0.928	0.920	

i) Precision:

The precision ratio is defined as the proportion of correctly projected positive observations to the total number of positive observations projected.

Precision = TP/TP+FP

ii) Recall:

Recall can be calculated by dividing the total number of True Positives in the data by the total number of positive class values. It is also referred to as the True Positive Rate or Sensitivity.

Recall = TP/TP+FN

i) F1 score:

The precision and recall weights are averaged to get the F1 Score. Consequently, false positives and false negatives are both considered in this score. Though F1 is generally more useful than accuracy, especially in cases where the distribution of classes is uneven, it is not intuitively as simple to understand as accuracy. When false positives and false negatives have comparable costs, accuracy performs best. It is preferable to consider both Precision and Recall if the cost of false positives and false negatives differs significantly.

F1 Score = 2*(Recall * Precision) / (Recall + Precision)

VI WORD PREDICTION

Word Prediction is to predict the next word in the sentence to complete the question. In our system, the technique is implemented using Keras in python. Keras is deep learning library that makes the deep learning simple. The work predictor workflow is depicted in the figure 2.

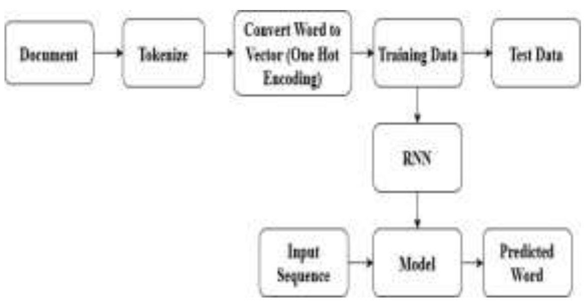

Figure – 2: Word Predictor workflow

RNN is used in our to build the model for predict the next word and Long Short-Term Memory (LSTM) plays a vital role to build the neural network using the softmax activation function in the RNN. The RNN used in our system shown in Figure 3.

To predict the next word in the sentence the input data are tokenized into different single words and the unique words are identified in the given data. Word to Vector is one of the techniques to prepare the data for training here One Hot Encoding used to convert the word into vectors. The One Hot Encoding representation is shown in Figure 4.

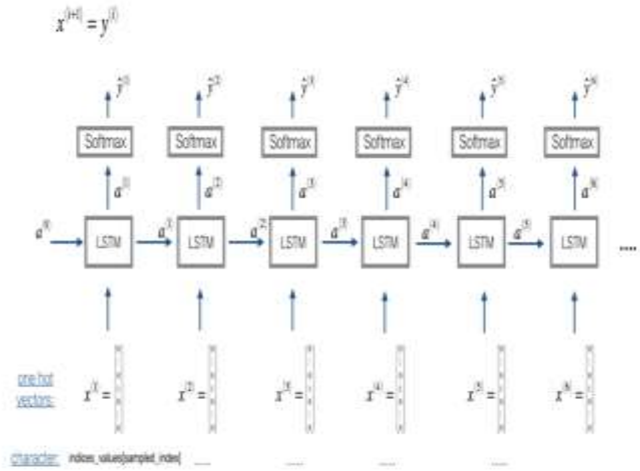

Figure – 3: Recurrent Neural Network.

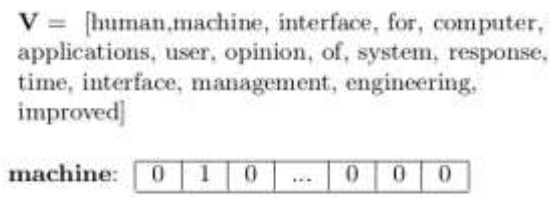

Figure – 4: One Hot Encoding

In the One Hot encoding representation, each word is represented with a large vector of size |V| i.e. vocabulary's size for the given corpus. The implementation of One Hot Encoding is very easy and easy to understand. This model predicts the next word using the previous sequence of five words.

VII CONCLUSION

The Automatic Question Generation system seeks to minimize human involvement while optimizing the likelihood of producing questions based on Bloom's Taxonomy cognitive levels. Only the English content is compatible with this system.

REFERENCES

[1] Robert M.Losse, Natural language processing in support of decision-making phrases and part-of-speech tagging, Information Processing & Management, Elsevier, 13 October 2000.

[2] Aravind K.Joshi, Natural Language Processing, University of Pennsylvania, Philadelphia, 13 SEPTEMBER 1991.

[3] Ming Liu, Rafael A. Calvo, and Vasile Rus, Automatic Question Generation for LiteratureReview Writing Support, Springer-Verlag Berlin Heidelberg 2010.

[4] Holy Lovenia, Agus Gunawan, Felix Limanta, Automatic Question-Answer Pairs Generation from Text, ResearchGate, November 2018.

[5] Karen Mazidi, Rodney D. Nielsen, Linguistic Considerations in Automatic Question Generation, Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, 2014 Association for Computational Linguistics.

[6] Vishwajeet Kumar, Nitish Joshi, Arijit Mukherjee, Ganesh Ramakrishnan, and Preethi Jyothi, Cross-Lingual Training for Automatic Question Generation, Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 4863–4872, 2019 Association for Computational Linguistics.

[7] Brain von konsky, David C.Gibson, Longwei Zheng, Parts of speech in Bloom's Taxonomy Classification, ResearchGate, November 2018.

[8] Qingyu Zhou, Nan Yang, Furu Wei, Chuanqi Tan, Hangbo Bao, and Ming Zhou, Neural Question Generation from Text: A Preliminary Study, Springer International Publishing AG 2018.