Automatic Question Generation using Natural Language Processing

***Synopsis Report submitted in partial fulfillment***

***of the requirement for the degree of***

**B. E.(Computer Engineering)**

Submitted By

ABHISHEK PHALAK

SHUBHANKAR YEVALE

GAURAV MULEY

Under the Guidance of

Prof. AMIT NERURKAR

Department of Computer Engineering

A picture containing qr code

Description automatically generated

Vidyalankar Institute of Technology

Wadala(E), Mumbai 400 037

## University of Mumbai

# 2021-22

### CERTIFICATE OF APPROVAL

**For**

**Project Synopsis**

This is to Certify that

ABHISHEK PHALAK

SHUBHANKAR YEVALE

GAURAV MULEY

Have successfully carried out Project Synopsis work entitled

Automatic Question Generation using Natural Language

Processing

in partial fulfillment of degree course in

Computer Engineering

As laid down by University of Mumbai during the academic year

2020-21

Under the Guidance of

Prof. AMIT NERURKAR

Signature of Guide Head of Department

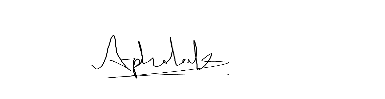
Examiner 1 Examiner 2 Principal

**<i>**

**Declaration**

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Name of student Roll No. Signature

1. Abhishek Phalak 18102A0027 

2. Shubhankar Yevale 18102A0031 

3. Gaurav Muley 18102A0038 

Date: 27/10/2021 **<ii>**

**Acknowledgements**

We wish to express our sincere gratitude to Prof. Amit Nerurkar for providing us an opportunity to

do our project on “Automatic Question Generation using NLP.” We sincerely thank Prof. Amit Nerurkar for his guidance and encouragement in carrying out this project work. We also wish to express our gratitude to other staff members of Vidyalankar Institute of Technology who rendered their help during the period of our project work. We also thank the Head of Department of Computer Engineering at Vidyalankar Institute of Technology, Dr. Sachin Bojewar for providing us an opportunity to embark on this project.

**<iii>**

**Table of Contents**

|  |  |
| --- | --- |
| **SR No.** | **Content** |
| 1 | Introduction |
| 2 | Aim and Objective |
| 3 | Literature Survey |
| 4 | Problem Statement |
| 5 | Scope |
| 6 | Proposed System |
| 7 | Methodology |
| 8 | Analysis  8.1 Process Module Used for the Project  8.2 Feasibility Study  8.3 Timeline Chart |
| 9 | Design - Dataflow Diagram |
| 10 | Hardware and Software Requirements |
| 11 | References |

**<iv>**

**Abstract**

Generation of questions from an extract is a very tedious task for humans and an even tougher one for machines. In Automatic Question Generation (AQG), it is extremely important to examine the ways in which this can be achieved with sufficient levels of accuracy and efficiency. The ways in which this can be taken ahead is by using the Natural Language Processing (NLP) to process the input and to work with it for AQG. Using the NLP with question generation algorithms the system can generate the questions for better understanding of the text document. The input is pre-processed before actually moving in for the question generation process. The questions formed are first checked for proper context satisfaction with the context of the input to avoid invalid or unanswerable question generation. This system can be used in various places to help ease the question generation and also at self-evaluator systems where the students can assess themselves so as to determine their concept understanding. This work can help improve the overall understanding of the level to which the concept given is understood by the candidate and the ways in which it can be understood more properly.

1. **Introduction: -**

The system accepts text documents as an input from the user, generates the questions based on various parts and parameters present in the text document itself. The system performs some pre-processing, finds the words/phrases that have the potential to be the answers for the questions that would be generated. These pivotal answers would then be used to form questions based on the context in which they are used in the document. The system forms syntactically and semantically correct questions from these answers, which are answerable in context to the document uploaded by the user. The questions which are important and answerable are further used to generate a self-evaluation quiz for the user to evaluate one’s understanding of the topic or concept.

1. **Aim and Objectives: -**

The aim of the project is to generate the questions in the form of a quiz, fill-in-the-blank, answer in one word, MCQ, crossword based on the document which is given as input to the computer so as to gauge the overall understanding about a chapter or a topic. The main motive of using automatic question generation systems is to generate simple and complex type questions to reduce the dependency on humans to generate questions and the time required to generate the reasonable questions.

**Objectives** –

1. To perform text analysis to extract the pivotal answer from the input sequences.
2. To automatically generate questions surrounding the pivotal answer.
3. To measure overall understanding of a topic using FIB, WH question and crosswords.
4. To conduct a test for the student using the generated questions.
5. To reduce the expenses associated with manual construction of questions and to satisfy the need for a continuous supply of new questions.
6. To have holistic approach in grading or assessing student’s grasp on the topic.
7. To reduce the expenses associated with manual construction of questions and to satisfy the need for a continuous supply of new questions.
8. The goal is to generate a valid and fluent question according to a given passage and the target answer.
9. **Literature Survey: -**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr.**  **No.** | **Author/Title/Year** | **Work done/Algorithm/Concept/Idea presented in the paper** | **Remarks** |
| 1 | **Putting the Horse Before the Cart: A**  **Generator-Evaluation**  **Framework for Question Generation from Text.**  Vishwajeet Kumar, Ganesh Ramakrishnan and Yuan-Fang Li  Published: Sept 2019 | * The paper aims at generating syntactically correct and semantically structured question generation from text or a knowledge base * The framework consists of a generator and evaluator. On generation of output by generator using a policy, the evaluator assigns a reward. Based on this reward, the generator then updates and improves the current policy * The Generator is a sequence-to-sequence model, augmented with   + an encoding for the potentially best pivotal answer,   + the copy mechanism to help generate contextually important words   + the coverage mechanism to discourage word repetitions | We can use this  architecture for generating the  questions. As it introduces better metrics to  evaluate the question, this might improve the efficiency of the question. |
|  |  | The evaluator ﬁne-tunes the parameters of the generator network by optimizing task-speciﬁc reward functions through policy gradient. The reward function can be calculated by metrics like BLEU, GLEU and ROUGE-L. Along with these, question generation specific metrics such as QSS, ANSS and DAS are also introduced for better performance |  |
| 2 | **Let’s Ask Again: Reﬁne Network for**  **Automatic Question Generation**  Preksha Nema, Akash Kumar Mohankumar, Mitesh M. Khapra, Balaji Vasan Srinivasan, Balaraman Ravindran.  Published: 2019 | It proposes Refine Network (RefNet), which examines the initially generated question and performs a second pass to generate a revised question. Furthermore, it proposes Reward-RefNet which uses explicit reward signals to achieve reﬁnement focused on speciﬁc properties of the question such as ﬂuency and answerability.  It analyses two models, the EAD model which contains single encoder and preliminary decoder And RefNet model which includes encoder, preliminary decoder and RefNet decoder. It is observed that the RefNet model performs better than EAD model  Based on analysis, RefNet rewards itself and then generates Refined questions.  The datasets used were SQUAD, HOTSPOT-QA and DROP and metrics included QBLEU, Bleu-n. | We can create a similar model that refines itself after the initial model has generated the question.  This will improve the accuracy of the models. |
| 3 | **Towards a Better Metric for Evaluating Question Generation Systems.**  Preksha Nema, Mitesh M. Kahpra  Published: 2018 | * The n-gram based similarity metrics do not correlate with human judgements in case of evaluating AQG and deem the unanswerable ones with the highest scores even though irrelevant. * The work on modifying the existing metrics to correlate with the human judgements. * These modifications are termed as Q-metrics modifying the existing metrics and naming them as Q-BLEU1 and so on. * Several datasets used for learning of this system are SQuAD, WikiMovies, VQA.   The new model of Reward based learning is used to evaluate the answerability of the questions. | The newly modified metrics prove to be very useful  in judging the  answerability of the  questions generated. |

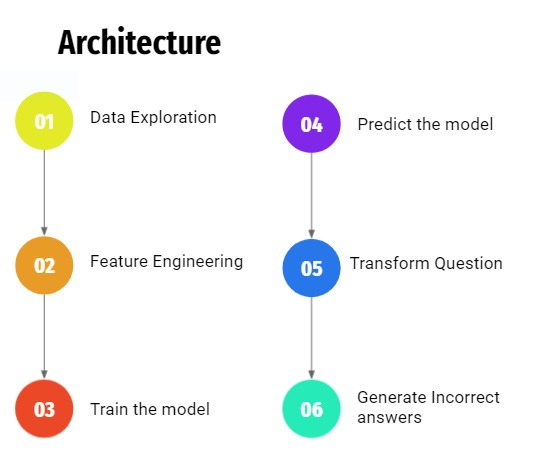
1. **Problem Statement: -**

To build a system that takes text document as input from user. The system should preprocess the input text, find most pivotal answers. It should generate questions, the answers to which are the pivotal answers generated in the previous step. Format these questions in the form of a quiz, fill-in-the-blank, answer in one word, MCQ, crossword to gauge the over understanding about a chapter or a topic.

1. **Scope: -**

The product aims at generating questions automatically. This is done by extracting the most pivotal answer from the input text. The question is generated surrounding this pivotal answer. The answerability and semantic correctness of the question is the main focus. Quiz is conducted for students and they can self-evaluate themselves.

1. **Proposed System: -**



After logging into the system the user has to upload a text document. The document cannot contain any images or non-textual special characters. The user is provided with the option of downloading or rating the generated question. Based on the choice, the generated question file or the ratings will be saved in the local disk.

Also, the user has to choose the maximum number of questions to be generated, although it cannot exceed the number of sentences in a text document. The questions are of two types, namely Fill in the blank (FIB) and Wh type of questions.

The FIB model uses machine learning techniques to generate FIB questions. It identifies the keyword using classification techniques. The keyword is then replaced with a blank line and the remaining sentence is used as the FIB question. It also generates multiple wrong answers or distractors using the keyword so that it can be used as MCQ. The Wh question first takes out the top sentences, preprocesses the text and then generates questions based on the type of sentences. Sentences are classified into NER tags based, discourse marker based and non discourse marker based algorithms. It applies various transformation rules to generate the questions based on the structure of the sentences.

**7) Methodology: -**

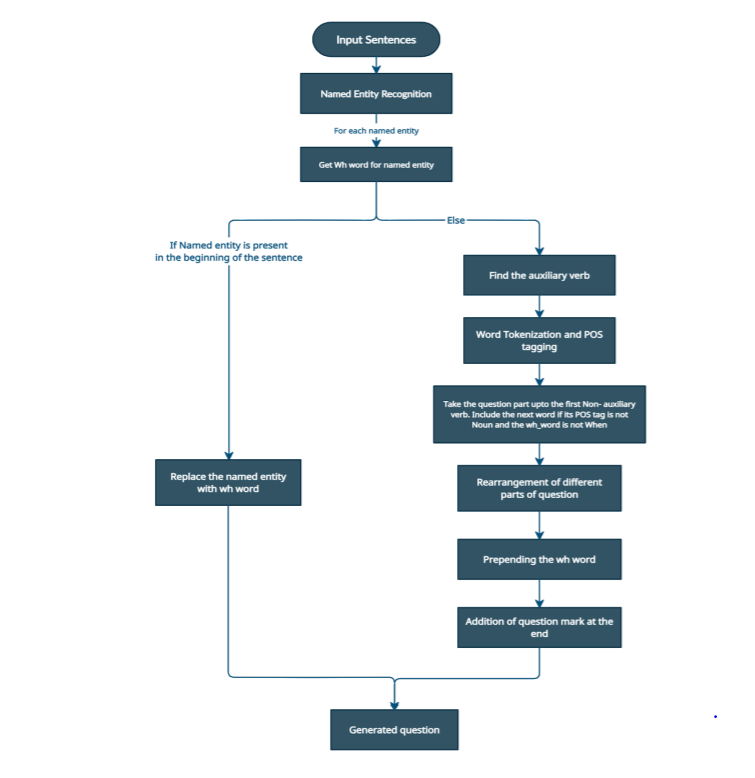
**FIB based questions**

We have used SQUAD 1.0 dataset which contains about 100,000 questions generated on Wikipedia articles. Intuitively, the task of selecting a probable answer is very much similar to tagging a word as spam or not spam. Hence, we decided to use binary classification on each input word to tag it whether it is an answer or not. For this task, each non-stop word from the paragraphs of SQUAD dataset were extracted and we added some features on them like POS tag, shape, word count, NER tag, etc. and a label ‘isAnswer’. Using the data generated from the previous step, we used scikit-learn's Gaussian Naive Bayes algorithm to train a model that would tag each word as whether it can be a pivotal answer or not. The advantage of using Naïve Bayes is that it also gives us the probability of each word, which will be used to choose the most probable pivotal answer. The distractors are generated using pre-trained word-embeddings and cosine-similarity. This will generate words that will be used as the multiple-choice options. Once the model is trained, we save the model to use it later for user inputs. After the user uploads the document. The content is split into sentences and various preprocessing is done to clean the text. We feed these sentences to the model that we saved earlier to predict the pivotal answers. The generated results are then formatted and displayed to the user.

**Wh- based Question**

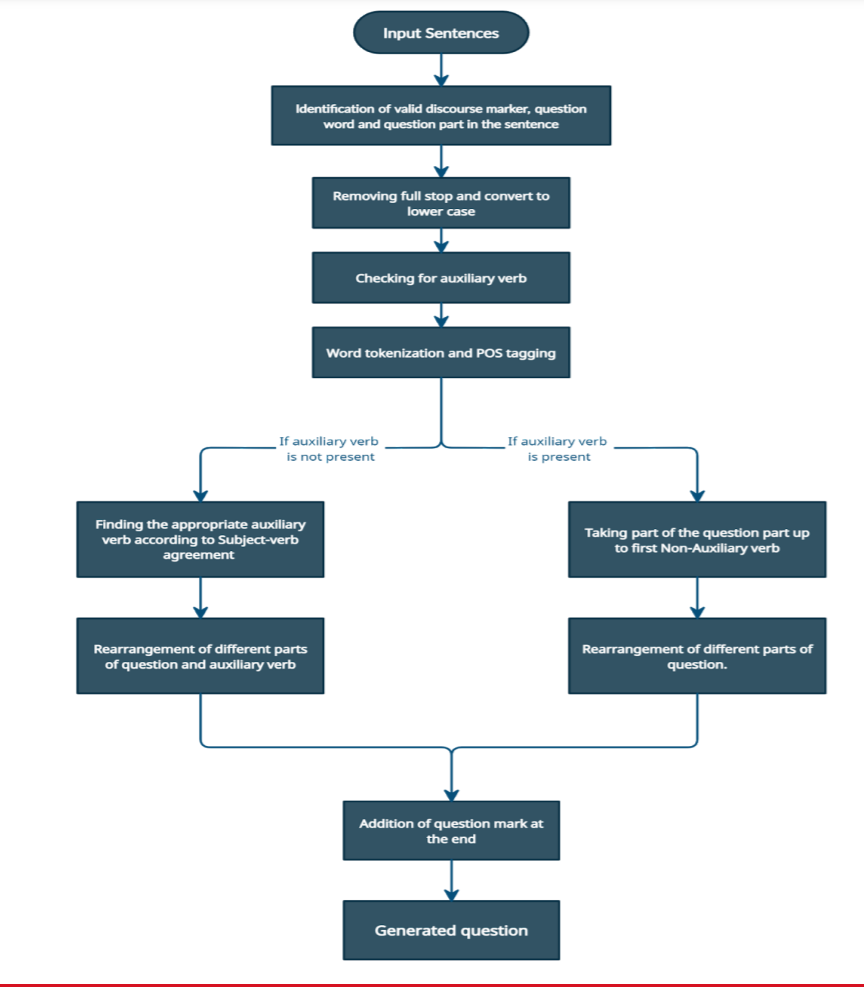
**1.Named Entity Recognition**

These are used in declarative or assertive sentences or Yes/no sentences. We use parts of speech (POS) tagging to find the nouns and then are run through Named Entity recognition to identify the type of named entity. Named entity recognition identifies these named entities in text and classifies them into predefined categories such as names of persons, organizations, locations, events, expression of times,numerical value, etc. Each of these named entities are mapped to a Wh word that would be used during question generation. For example if the named entity is a name, then the appropriate question word would be Who. Similarly if the named entity is time, the question word would be When. After identification of the named entity we have made rules to generate the question. According to the rules, if the named entity is present in the beginning of the sentence, we can simply replace the named entity with the appropriate Wh word to form the question. We find the auxiliary verb in the sentence if the named entity is not present in the beginning. Iterating over the POS tags, we append each word into the question part until we find the first non auxiliary verb. If the word after this non-auxiliary verb is a noun ,we discard it. If it is not a noun, we include it in the question part. We then rearrange the question parts satisfying the grammatical syntax. After rearrangement the Wh word is prepended and the question mark is added at the end to generate the whole question



**Discourse Marker based Question**

In this type of question generation, we first identify discourse markers like ‘because’, ‘although’ etc. For every discourse marker present in the sentence we define a question word associated with it. Then we process the sentence by converting it into lowercase and removing full stop. We further classify the sentence into two types, one having an auxiliary verb and ‌ one without an auxiliary verb. Using the POS and NER tags generated in the general preprocessing part, we identify :



If sentence has an auxiliary verb:

● We append each word into the question part until we find the first non-auxiliary verb in the sentence.

● Split the question part across the auxiliary verb and place the auxiliary verb at the start of the question part.

● If the question type is of Yes/no, we return the formed question part with a question mark appended, else prepend the question word at the start of the question part and return.

If the sentence does not contain any auxiliary verb :

● We define various combinational rules to generate an appropriate auxiliary verb. We identify which verb will be appropriate to fulfill the subject verb agreement. The auxiliary verbs like do, does and did may be included in the question part

● We stem the verb present in the sentence using any stemmer or a lemmatizer

● Rearrange the question word so as to fulfill all the grammatical rules

● Prepend the question word, as identified the discourse marker’s mapping

● Append the question mark and return the generated question

**8) Analysis**

**8.1 Processed Model Used for the Project**

For fill in the blank questions, the squad v1 dataset is preprocessed and various functions of nltk library were used like POS tagging, NER tagging etc. to create features of the dataset. The preprocessed data is then trained on machine learning model of Naïve Gausian Nb.

**8.2 Feasibility**

We looked at a lot of similar applications in the market which were based on complex neural

network models and which used huge datasets to go about the task of question generation.

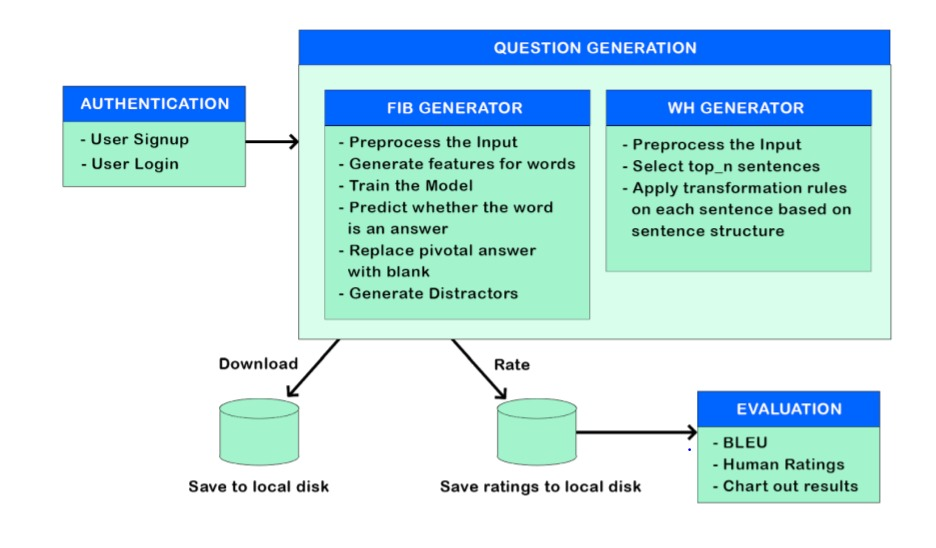
Hence acquiring a dataset which would have passage-question-answer pairs would be the first concern for the system. This would be solved by using SQUAD 1.0 Dataset which has over 100,000 such pairs on wikipedia articles.

**8.3 Time Line Chart**

|  |  |  |
| --- | --- | --- |
| **Sr.**  **No.** | **Date** | **Project Work Done / Progress Achieved** |
| 1 | 12/07/21  to 27/07/21 | Searching for a problem and finding existing solutions for the problem |
| 2 | 29/07/21  to 08/08/21 | Laid down a rough solution to the problem of question generation, looking for various methods for AQG task |
| 3 | 13/08/21  to 25/08/21 | Started with literature survey and paper reading |
| 4 | 28/08/21  to 18/09/21 | Studied different methodologies discussed in the papers and sorted the ones fit for our project |
| 5 | 20/09/21  to 26/09/21 | Studied the technologies used in the papers for better understanding of the workings |
| 6 | 28/09/21  to 04/10/21 | Searched for datasets that will be used for training |
| 7 | 05/10/21  to 12/10/21 | Designed the overall architecture of the system |
| 8 | 14/10/21  to 20/10/21 | Built Machine Learning Model for Wh type Question |
| 9 | 20/10/21  to 23/10/21 | Preparing the presentation and report for final review |

**9) Design**

**Dataflow Diagram**

****

**10) Hardware and Software Requirements: -**

**Technology Stack**-

* HTML
* CSS
* JavaScript
* Bootstrap
* Flask
* Python
* Scikit Learn
* Matplotlib
* Pandas
* Numpy
* NLTK
* Tensorflow
* Keras.

**11) References: -**

[1] Michael Heilman and Noah A. Smith. (2010). Good question! statistical ranking for question generation. In HLT-NAACL, pages 609–617. The Association for Computational Linguistics.

[2] Pranav Rajpurkar ,JianZhang, Konstantin Lopyrev, and Percy Liang. (2016). Squad: 100, 000+ questions for machine comprehension of text. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas,USA, November1-4,2016, pages 2383–2392.

[3] Kishore Papineni, Salim Roukos, Todd Ward, and WeiJing Zhu. (2002). Bleu: a method for automatic evaluation of machine translation. In ACL, pages 311– 318. ACL.

[4] Bloom, B.S. (1956). Taxonomy of educational objectives: The classification of educational goals.

[5] Chin-Yew Lin. (2004). Rouge: A package for automatic evaluation of summaries. Text Summarization Branches Out.

[6] Michael Denkowski and Alon Lavie, "Meteor Universal: Language Specific Translation Evaluation for Any Target Language", Proceedings of the EACL 2014 Workshop on Statistical Machine Translation, (2014).

[7] Vishwajeet Kumar, Kireeti Boorla, Yogesh Meena, Ganesh Ramakrishnan, and Yuan-Fang Li. (2018). Automating reading comprehension by generating question and answer pairs. In 22nd Pacific-Asia Conference on Knowledge Discovery and Data Mining.

[8] Vishwajeet Kumar, Yuncheng Hua, Ganesh Ramakrishnan, Guilin Qi, Lianli Gao, and Yuan-Fang Li. (2019) Difficult-controllable multi-hop question generation from knowledge graphs.

[9] Serban, I., Sordoni, A., Bengio, Y., Courville, A., & Pineau, J. (2016). Building End-To-End Dialogue Systems Using Generative Hierarchical Neural Network Models. Proceedings of the AAAI Conference on Artificial Intelligence.

[10] Preksha Nema, Akash Kumar Mohankumar, Mitesh M. Khapra, Balaji Vasan Srinivasan, Balaraman Ravindran , Let's Ask Again: Refine Network for Automatic Question Generation, EMNLP (2019).

[11] Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. (2015). Pointer networks. In Advances in Neural Information Processing Systems, pages 2692–2700.

[12] Nan Duan,Duyu Tang, Peng Chen, Ming Zhou, Proceedings of the (2017) Conference on Empirical Methods in Natural Language Processing, pages 866–874.

[13] Chin-Yew Lin. (2004). Rouge: A package for automatic evaluation of summaries. In Proc .ACL workshop on Text Summarization Branches Out, page 10.

[14] Roger Azevedo, Amy Witherspoon, Arthur Graesser, Danielle McNamara, Amber Chauncey, Emily Siler, Zhiquiang Cai, Vasile Rus, Mihai Lintean, (2009) Analyzing Self-Regulated Learning in a Tutoring System for Biology, Frontiers in Artificial Intelligence and Applications, pages 635 - 637

[15] Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK Li. (2016). Incorporating copying mechanism in sequence-to-sequence learning. In ACL, volume 1, pages 1631–1640

[16] Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. (2016). Modeling coverage for neural machine translation. In ACL 2016, pages 76–85. The Association for Computer Linguistics.

[17] Yao Zhao, Xiaochuan Ni, Yuanyuan Ding, and Qifa Ke. (2018). Paragraph-level neural question generation with maxout pointer and gated self-attention networks.

[18] Q. Luo, B. Liu, J. Yan and Z. He, "Research of a Spam Filtering Algorithm Based on Naïve Bayes and AIS," 2010 International Conference on Computational and Information Sciences, (2010), pp. 152-155.

[19] Rish, Irina. (2001). An Empirical Study of the Naïve Bayes Classifier. IJCAI 2001 Work Empir Methods Artif Intell. 3.

[20] Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean, “Efficient Estimation of Word Representations in Vector Space”, (2013)

[21] Rahutomo, Faisal & Kitasuka, Teruaki & Aritsugi, Masayoshi. (2012). Semantic Cosine Similarity.

[22] Prashanth Mannem, Rashmi Prasad and Aravind Joshi. (2010). Question Generation from Paragraphs at UPenn: QGSTEC System Description, Proceedings of QG2010: The Third Workshop on Question Generation