Automation of Question-Answer Generation

Alok Kumar
Dept. of Computer Science and Engg.
Chhatrapati ShahuJi Maharaj University
Kanpur, India
akumar.uiet@gmail.com

Deepika Singh
Dept. of Computer Science and Engg.
Chhatrapati ShahuJi Maharaj University
Kanpur, India
ds.deepikasingh19@gmail.com

Aditi Kharadi
Dept. of Computer Science and Engg.
Chhatrapati ShahuJi Maharaj University
Kanpur, India
aditikharadi26@gmail.com

Mala Kumari
Dept. of Computer Science and Engg.
Chhatrapati ShahuJi Maharaj University
Kanpur, India
km.malakumari.12@gmail.com

***Abstract*—In this paper, an automatic system is proposed to generate different kinds of questions and answers from the input text. Question answer generation systems have been an interesting field of research for over decades. From generating questions for educational purposes to preparing answers to questions that could be asked in a legal proceeding, the purpose of question answer generation(QAG) systems is to reduce the tedious task of going through large texts. In our system, question-answer pairs from a given input text are generated using linguistic and statistical knowledge of text. Initially, those sentences are identified from the input text on which questions can be framed and in further steps, identified sentences are ranked in an order of importance. Built specifically for assessment purpose, the system generates multiple choices, fill-ups, true-false, binary and wh type questions based on high ranked sentences in the last step. In performance evaluation of the proposed system, it is found that the system is performing well and it is close to state of art research in the standard linguistic approach.**

***Index Terms*—Named Entity Recognition(NER), POS tagging(parts of speech), NLTK(Natural Language Toolkit), Natural Language Generation(NLG), Dependency Parsing.**

I. Introduction

This system is built specifically for examination purposes. Examiners, in general, deal with the tedious task of generating questions and answers for testing the knowledge of students. This process of assessment has proven to be efficient for increasing knowledge and interest of students in studies. A general assessment may contain various types of questions like Wh (Who, What, Where, How), Fill ups, MCQs, True-False, etc. The examiner's main goal is to test the student's knowledge over certain important aspects of a topic. Besides other human errors, the possibilities of missing some important concepts while forming the questions exist too. Hence the need of an automatic question answer generator arises.

Our system uses natural language processing to generate possible questions out of the text provided as input. Natural Language processing makes human language intelligible to machines by combining linguistics and computer science. It is used to understand the structures and semantics of a human language by analysing its syntax, pragmatics and morphology and transforming this understanding into machine learning algorithms. Human-understandable language can be converted to machine-understandable language through NLU (Natural Language Understanding), which is essential for our system. Furthermore, we need to determine the content of the output that our system generates, which is a task of Natural Language Generation(NLG). Lastly, we need to produce linguistically valid results. In our system, the questions are framed on a list of sentences sorted out of the corpora. The answer (or gap) is selected from a potential key list formed by filtering each sentence for either of the noun, superlative or a named entity. The system then frames the questions on the selected sentences using a rule based approach.

II. Related Work

Many researchers have proposed various methods for generating question-answer pairs from text. There are two possible approaches for the task - NLP and Deep NLP (or Deep Learning).

The initial QAG systems have used rule-based approaches as proposed in papers by (Heilman and Smith, 2011 [18]; Lindberg et al., 2013 [15]; Labutov et al., 2015 [16]). Lovenia et al. [1] proposed usage of various text summarization algorithms to obtain important sentences, defining a custom NER for gap selection followed by a rule-based Wh-Question formation. They promote usage of TextRank, Multi word Phrase Extraction and Latent Semantic Analysis to compute the importance of sentences. Another similar approach by Nibras et al. [2] uses NER and POS for obtaining noun phrases(NP) and exploiting them as key phrases for framing the questions. For each declarative sentence, all the unmovable phrases (non- answer phrases) are marked. Furthermore, the questions are framed on answer phrases by replacing them by question phrases after decomposing the main verb, inverting the auxiliary and subject verb. Post-processing is performed to ensure proper formatting of the questions. The answers are then extracted from the framed questions using cosine similarity and pattern extraction.

A paper by Laddha et al. [3] suggests selection of descriptive sentences and forming a potential key list out of it.

Furthermore, the answer list is formed of phrases (which are essentially nouns, superlatives or named entities) that occur the least number of times. The answers are then selected as gaps and distractors are chosen from amongst antonyms and synonyms. The NLG systems include validation of output in order to avoid linguistic and semantic errors.

Deep learning has been found to be a promising approach for the NLP tasks. Sequential models like Google's T5 and BERT have been the new benchmarks for tasks like ours. Various papers on QAG have proposed the use of this approach. Dong Bok Lee et al. [7] propose the use of a hierarchical conditional variational autoencoder (HCVAE) with the goal of generating diverse and consistent question answer QA pairs. For given unstructured texts as contexts, the mutual information between generated QA pairs is maximized to ensure their consistency [7]. For the resolving consistency issue between answers and questions, use of Infomax regularizer is proposed which increases the mutual information between them.

Han Xiao et al. [8] introduce a two-way sequence transduction model that inculcates three modalities(question, answer and context)[8]. Other methods include usage of knowledge graphs like Freebase(Sathish Indurthi et al. [9]), neural encoder-decoder models(Zhou et al. , 2017 [12]), fine tuning of publicly available models like BERT(Devlin et al., 2018 [10]) and fine tuned QG-AG models with roundtrip consistency(Alberti et al., 2019 [11])

III. Proposed Methodology

Techniques used in question generation can be broadly described in following steps:

1) **Preprocessing**: input text is sentence tokenized, converted into lowercase, punctuation marks and stopwords are removed.
2) **Sentence Selection**: Select the sentence containing NER tag or POS tag (Noun, Pronoun and Superlative Degree).
3) **Text Ranking**: Rank all the selected sentences using TextRank Algorithm.
4) **Gap Selection**: Choose appropriate blank space in the sentence.
5) **Alternative Choice Selection**: Draft three choices which have the same contextual meaning as the selected gap.
6) **Question Formation**: Display the questions based on the rank of sentences using POS tags, named entities and sentence matter extraction (subject, verb, object, tense and prepositional phrase).

The flow of above mentioned processes is shown in Fig. 1.

A. Preprocessing

The input passage is tokenized into sentences followed by conversion into lowercase, removal of punctuation marks and stopwords.

B. Sentence Selection

All the sentences present in the passage might not be useful for question formation. So, to reduce our system's processing time, we eliminate such sentences. Such sentences are the ones which do not have any named entity, noun or superlative degree present in them or the ones that are too small for question formation (length less than three).Example:

- He went there.
- let me try to summarize it for you.
- Go over there.

Finally, the sentences that do not fall in the above defined criteria are selected for the question formation process.

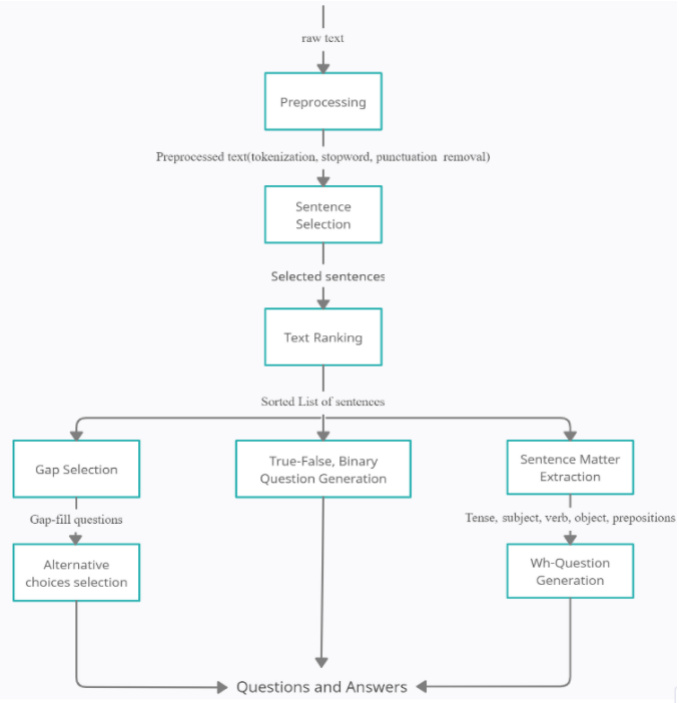

Fig. 1. Process Flow

C. Text Ranking

All the selected sentences might not be in the order of their importance for the question generation. So, the list of selected sentences is sorted using the Text-Rank Algorithm.

TextRank For the selection of important sentences, we need to rank them first. We propose to use the TextRank algorithm. It is an extractive text summarization method which ranks the sentences on the assumption that most important sentences are the ones that are most similar to other sentences [1]. There are numerous methods to find the similarity, for example, jaccard distance, cosine distance, etc. The similarity between the sentences has been computed using cosine distance. Below are the steps involved:

1) **Sentence Embedding**: The sentences are converted to vector representations using Word2Vec.

2) **Similarity Matrix**: The cosine similarity is the measure of similarity between two given vectors of an Hausdorff

pre-Hilbert space. The formula for finding cosine similarity between vectors A and B is shown in Eq.1

$$\cos(\theta) = \frac{A.B}{||A||\,||B||} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \quad ...(1)$$

where Ai , Bi are the components of these vectors. For each pair of sentences, the cosine similarity is computed and stored in a matrix called similarity matrix. This similarity matrix is normalized before further processing.

3) **PageRank Algorithm**: The PageRank algorithm was introduced to rank web pages in Google search. It is a graph based algorithm that outputs a probability distribution indicating the likelihood of a user arriving at a certain page, where each page is considered as a node of the graph with outbound links to other pages as edges. The formula for calculation of the ranks is shown in Eq. 2.

$$PR(A) = \frac{1-d}{N} + d\left(\frac{PR(B)}{L(B)} + \frac{PR(C)}{L(C)} + ..\right) \quad ...(2)$$

Where d is the damping factor, PR(X) is the page rank of the web page X, N is the number of web pages and L(Y) is the number of outbound links from node Y. PageRank algorithm is run iteratively until convergence to achieve good results. This mechanism is used to rank the sentences.

4) **Sentence Ranking**: Top-N sentences from the list (sorted on the basis of page ranks) are selected as important sentences.

D. Gap Selection

After getting the final list of selected sentences, the next important step is selecting the blank space. If there is any named entity present in the sentence, it is directly selected as the blank space and question is formed. Otherwise a potential key list is formed pushing all the noun, pronoun and superlative degree in the sentence. The best key is chosen on the basis of occurrence of key present in the potential key list, in the entire context. i.e., the key occurring for the greatest number of times is chosen as our best key, which is further used for framing questions. In order to obtain better results, we fine tuned the NER's performance by customizing and storing the unrecognized/wrongly recognized entity-label pairs separately for later use. Several NERs can be customized as required, for example, Spacy provides pre-trained NER models that can be fine tuned or even built from scratch. While testing our custom NER for a domain, we found that the model was forgetting the elements learnt in pretraining, and its performance worsened, the reason being **Catastrophic Forgetting**. Before training the model, one has to ensure that ample pre training examples are included in the training data in order to facilitate its improvement. Our system is not bound to a single domain and training a custom NER that works for all domains would be a highly complex task. Hence, we improve the system's performance by customizing NERs as per domain requirements.

E. Alternative Choice Selection

After selecting the gap, we have the correct answer for our question. Next step is to find three alternative choices which will distract the learner. For generating the distractors, we use the Sense2vec python library. In this library the relation between words is generated automatically using text corpus. Relationship between words is found by representing each word as an array (word vectors). There are cases when Sense2vec is not able to generate suitable distractors. We use wordnet in such cases. In wordnet, a word may have different senses. Example: "Apple" may have different senses, one for the multinational company and other as a fruit. When mentioning complete sentences, algorithms are not good at guessing the sense of the word, unlike us. Here we have used the topmost sense for generating the distractors. After getting all the possible distractors, we randomly choose three of them and display it with our correct answer (i.e., the blank chosen).

F. Question Formation

After the distractors are selected, we use nouns, superlatives and named entities as gaps or answer phrases, based on which, we form following types of questions:

1) Multiple choice blank questions.
2) Fill in the blank questions
3) Binary Questions
4) True False Questions
5) Wh-Questions

1) **Multiple Choice Blank Questions** Finally the selected blank key is replaced with a blank space in the sentence. The correct answer and selected distractors are arranged randomly to get the final choices. Then the sentence and its respective choices are displayed as output in decreasing order of rank of each sentence.
2) **Fill in the blanks Questions** Fill up questions are formed in a similar way as described above. The named entities are used as blanks for the fill up questions. For each named entity, the corresponding position is replaced by a blank.
3) **Binary Questions** For this, we propose to exploit auxiliary verbs. For forming more questions, we can negate the auxiliary verb. For example, 'was' becomes 'was not'. Furthermore, we can transform the sentence into interrogative form by shifting the auxiliary verb to the beginning of the sentence. E.g., 'Elon Musk is an intelligent person.' becomes 'Is Elon Musk an intelligent person?' and 'Is Elon Musk not an intelligent person?'. Besides this, we have also included the results obtained from NLG (elaborated in 5).
4) **True False Questions** True-false questions are essentially statements or general sentences. Similar to the

Binary Questions generation, we can use both the auxiliary verb and its complement form to generate more questions. For example, 'Obama was a good president.' can be converted into 'Obama was not a good president.'.

5) **Wh-Questions** The Wh-question generation uses a rule based approach. Sentence matter extraction is crucial for this task. We extract the verb in a sentence, followed by its subject, object, prepositional phrase and tense form. Question Generation is a task of NLG, which helps transform the structured data into natural language.The NLG task may include various steps like determining content, structuring the document, aggregation, lexical choices, generating referring expressions and finally, realization. The system uses named entity recognition, parts of speech tagging and dependency parsing to retrieve subject, verb, object, prepositional phrase and tense of a sentence.Simplenlg is a library that utilizes this information for question generation. We use simplenlg to generate WH-questions as per the below rules:
   a) If the answer phrase is a subject or object, we simply convert the simplenlg object to its interrogative form, using the NER tag for question word selection.
   b) If the answer is a prepositional phrase, we generate a simplenlg object without adding the complement. An appropriate question word is then chosen, based on the named entity tag of the preposition phrase.
   c) If the answer phrase contains an adverb, we transform the phrase to its interrogative form, using 'How' as the question word.

IV. Evaluation

Evaluation of the system is done on the basis of the number of QA pairs formed and their correctness for a given context. In order to determine the fraction of relevant instances produced by the system, recall is used, which is calculated using Eq. 3. Precision is a measure that gives the correctness of the QAG system and we calculate precision using Eq. 4. One way of finding the overall accuracy is F1 score. F1 score is calculated using Eq. 5. For the purpose of evaluation, we compared the output generated for 200 passages of variable length with expected output, which was collected through a survey. Table I shows the scores obtained by the system and the plot of recall, precision and F1 score is presented in Fig. 2.

$$Recall = \frac{No.\ of\ QA\ pairs\ generated}{Expected\ no.\ of\ QA\ pairs} \quad ...(3)$$

$$Precision = \frac{Correct\ QA\ pairs\ generated}{Total\ QA\ pairs\ generated} \quad ...(4)$$

$$F1Score = \frac{2 * Recall * Precision}{Recall + Precision} \quad ...(5)$$

TABLE I
System Performance

Type	Recall	Precision	F1 score
MCQs	0.79	0.90	0.84
Fill-ups	0.93	0.97	0.95
True-False	0.98	0.76	0.86
Binary	0.91	0.80	0.84
Wh	0.83	0.73	0.77

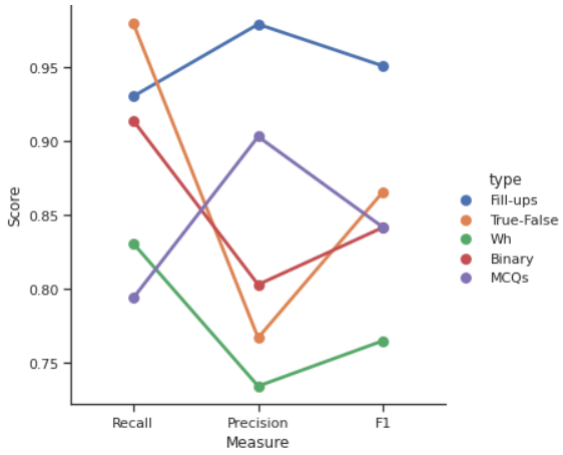

Fig. 2. A graph of scores obtained.

V. Result

Our system successfully generates MCQs, Gap filling, True-False and Binary type questions on the given sentences. For the Wh-question generation, the system performs well for direct simple sentences, but falters on the complex sentences. We have used both spacy and nltk for entity recognition in order to obtain better results. For the question generation system, we made following assumptions:

1) Generally, the answer to the gap-filling questions are named entities. This might not hold for all scenarios like English Language, but is convincing in case of other contexts.
2) The sentences provided to the Wh-Question generation system are simple declarative sentences, which can be used to form the questions directly.

Below is a sample set of question-answer pairs from output generated by our system for two different sentences taken from Wikipedia articles:

1) *'"Apple's software includes macOS, iOS, iPadOS, watchOS, and tvOS operating systems, the iTunes media player, the Safari web browser, the Shazam music identifier and the iLife and iWork creativity and productivity suites, as well as professional applications like Final Cut Pro, Logic Pro, and Xcode. '"* [24]
   a) _____*'s software includes macOS, iOS, iPadOS, watchOS, and tvOS operating systems, the iTunes media player, the Safari web browser, the Shazam music identifier and the iLife and iWork*

creativity and productivity suites, as well as professional applications like Final Cut Pro, Logic Pro, and Xcode.
Ans: Apple

b) *Apple's software includes macOS, iOS, iPadOS, watchOS, and tvOS operating systems, the iTunes media player, the Safari web browser, the _____ music identifier and the iLife and iWork creativity and productivity suites, as well as professional applications like Final Cut Pro, Logic Pro, and Xcode.*
Ans: Shazam

c) *What does Apple's software include?*
Ans: Apple's software includes macOS, iOS, iPadOS, watchOS, and tvOS operating systems, the iTunes media player, the Safari web browser, the Shazam music identifier and the iLife and iWork creativity and productivity suites, as well as professional applications like Final Cut Pro, Logic Pro, and Xcode.

2) *'"A computer is a machine that can be instructed to carry out sequences of arithmetic or logical operations automatically via computer programming."' [25]*

a) *A computer is a _____ that can be instructed to carry out sequences of arithmetic or logical operations automatically via computer programming*
a- bag
b- machine
c- pipe
d- homemade
Ans: machine

b) *Is a computer a machine that can be instructed to carry out sequences of arithmetic or logical operations automatically via computer programming?*
Ans: Yes

c) *Is a computer not a machine that can be instructed to carry out sequences of arithmetic or logical operations automatically via computer programming?*
Ans: No

d) *A computer is not a machine that can be instructed to carry out sequences of arithmetic or logical operations automatically via computer programming.*
Ans: False

e) *What is a machine, and carry of arithmetic logical operations via computer programming?*
Ans: A computer is a machine that can be instructed to carry out sequences of arithmetic or logical operations automatically via computer programming.

In the first question, legit question-answer pairs are generated for gap-filling and Wh-questions. The second sentence is a declarative sentence with no named entities. The questions generated for this sentence are MCQs, Binary, True-False and Wh-type. The gap-filling questions are not generated by our system for a sentence with no named entity. As per the Wh-question generated on the second sentence, our system falters for sentences with multiple verbs.

VI. Conclusion and Future Work

Using the standard linguistic approach, we were able to generate questions from text. We used parts of speech tagging, named entity recognition, dependency parsing to retrieve sentence matter and form various questions. We improved our system's performance by fine tuning the named entity recognition in the system. The rule based approach for Wh-question generation may falter for compound and complex sentences. While exploring question generation, we came across limitations of a rule-based system. This is because english sentences can be ambiguous at times. Besides this, while our system performs well at framing questions on the sentences of an input paragraph, it misses on the connection between the sentences. The future work includes using even firmer rules and deeper analysis of the paragraph in the standard linguistic approach for better performance. Another approach could be using deep neural networks or transformers for making the Wh-question generation system more robust.

References

[1] Lovenia, H., Limanta, F., Gunawan, A, "Automatic Question-Answer Pairs Generation from Text," Technical report, Bandung Institute of Technology. 2018.

[2] A.S.M Nibras, M.F.F Mohamed, I.S.M Arham, A.M.M Mafaris and M.P.A.W Gamage, Automatic Question and Answer Generation from Course Materials. International Journal of Scientific and Research Publications, Volume-7, Issue-11. 2017.

[3] Miss.Pranita Pradip Jadhav, Mrs.Manjushree D. Laddha. Automatic Gap-filling question generation. International Journal of Computer Science Engineering Technology, Volume-8, Issue-8. 2017.

[4] Manish Agarwal and Prashanth Mannem. Automatic Gap-fill Question Generation from Text Books. Association for Computational Linguistics. 2011.

[5] Saidalavi Kalady, Ajeesh Elikkottil and Rajarshi Das. Natural Language Question Generation using Syntax and Keywords. Proceedings of QG2010: The Third Workshop on Question Generation. Volume-2, Pages 5-14. 2010.

[6] Kaksha Mhatre Akshada Thube Hemraj Mahadeshwar and Prof.Avinash Shrivas. Question Generation using NLP. International Journal of Scientific Research Engineering Trends Volume-5, Issue-2. 2019

[7] Dong Bok Lee, Annie Lee, Won Tae Jeong, Dong Hwan Kim and Sung Ju Hwang. Generating Diverse and Consistent QA pairs from Contexts with Information-Maximizing Hierarchical Conditional VAEs. The 58th Annual Meeting of the Association for Computational Linguistics, Pages 208-224. 2020

[8] Han Xiao, Feng Wang, Jianfeng Yan and Jingyao Zheng. Dual Ask-Answer Network for Machine Reading Comprehension. arXiv:1809.01997v2 . 2018

[9] Sathish Indurthi, Dinesh Raghu, Mitesh M. Khapra and Sachindra Joshi. Generating Natural Language Question-Answer Pairs from a Knowledge Graph Using a RNN Based Question Generation Model. Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume-1. 2017

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume-1, Pages 4171-4186. 2019

[11] Chris Alberti, Daniel Andor, Emily Pitler, Jacob Devlin and Michael Collins. Synthetic QA Corpora Generation with Roundtrip Consistency. arXiv:1906.05416 .2019

[12] Qingyu Zhou, Nan Yang, Furu Wei, Chuanqi Tan, Hangbo Bao and Ming Zhou. Neural Question Generation from Text: A Preliminary Study. arXiv:1704.01792 . 2017

[13] Zi Chai and Xiaojun Wan, Learning to Ask More: Semi-Autoregressive Sequential Question Generation under Dual-Graph Interaction,Volume: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Pages 225-237. 2020.

[14] Michael Heilman and Noah A. Smith, Good question! statistical ranking for question generation. Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Pages 609-617. 2010.

[15] David Lindberg, Fred Popowich, John C. Nesbit, and Philip H.Winne, Generating natural language questions to support learning on-line, Volume-Proceedings of the 14th European Workshop on Natural Language Generation, Association for Computational Linguistics,Pages 105-114. 2013

[16] Igor Labutov, Sumit Basu, and Lucy Vanderwende, Deep questions without deep understanding, Volume-Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, Association for Computational Linguistics, Pages: 889-898. 2015.

[17] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Matthew Kelcey, Jacob Devlin, Kenton Lee, Kristina N. Toutanova, Llion Jones, Ming-Wei Chang, Andrew Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov . Natural questions: a benchmark for question answering research,volume -8,Transactions of the Association for Computational Linguistics. 2019.

[18] Michael Heilman,. Automatic factual question generation from text.International Conference on Innovation in Intelligent Systems and Application. 2011.

[19] Anoop Kunchukuttan, Pratik Mehta, and Pushpak Bhattacharyya, The IIT bombay english-hindi parallel corpus, Volume-Proceedings of the Eleventh International Conference on Language Resources and Evaluation, European Language Resources Association. 2018.

[20] Duyu Tang, Nan Duan, Tao Qin, Zhao Yan, and Ming Zhou, Question answering and question generation as dual tasks,arXiv preprint arXiv:1706.02027, Microsoft Research Asia, Beijing, China Beihang University, Beijing. arXiv:1706.02027. 2017.

[21] Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. A Survey on Deep Learning for Named Entity Recognition. arXiv:1812.09449v3,IEEE Transaction on Knowledge and Data Engineering, PP(99). 2020.

[22] Amit Mandelbaum and Adi Shalev. Word Embeddings and Their Use In Sentence Classification Tasks. arXiv: 1610.08229v1. Hebrew University of Jerusalem.2016.

[23] Aparna Bulusu and Sucharita V. Research on Machine Learning Techniques for POS Tagging in NLP. Volume-8, Issue-1S4 . International Journal of Recent Technology and Engineering. 2019.

[24] Wikipedia contributors. Apple Inc. Wikipedia, The Free Encyclopedia. 2020 Dec 14, 12:26 UTC. Available here .

[25] Wikipedia contributors. Computer. Wikipedia, The Free Encyclopedia. 2020 November 9, 22:41 UTC. Available here.