A Question Answering and Quiz Generation Chatbot for Education

Sreelakshmi A.S.
Department of Computer Science and Engineering
National Institute of Technology, Tiruchirappalli
Tamil Nadu, India
sreelakshmias97@gmail.com

Abhinaya S.B.
Department of Computer Science and Engineering
National Institute of Technology, Tiruchirappalli
Tamil Nadu, India
sbabhinaya@gmail.com

Aishwarya Nair
Department of Computer Science and Engineering
National Institute of Technology, Tiruchirappalli
Tamil Nadu, India
aishwarya05nair@gmail.com

Jaya Nirmala S.
Department of Computer Science and Engineering
National Institute of Technology, Tiruchirappalli
Tamil Nadu, India
sjaya@nitt.edu

***Abstract*—In recent years, there have been a number of chatbots developed in the field of education. While many of them are designed to answer queries based on publicly available information such as in community answering platforms, or from a predefined knowledge base, there is no possibility of customizing the information to be queried. Moreover, there are no existing chatbots capable of generating self assessment quizzes based on any given document. This paper proposes a Question Answering and Quiz Generation Chatbot that allows a user to upload relevant documents and perform two main functions on them, namely answer extraction and question generation. The uploaded document is converted into a knowledge base through a number of data cleaning and preprocessing steps. The Question Answering module uses ranking functions and neural networks to extract the most appropriate answer from the knowledge base and the Quiz Generation module identifies key sentences and generates question-answer pairs, which can be used to generate a quiz for the user.**

***Index Terms*—Question Answering, Question Generation, Artificial Neural Networks (ANN), Text ranking, MS MARCO dataset, Stanford Coref Annotator.**

I. Introduction

A chatbot is a software used to imitate human conversation through text chats and voice commands. In recent times, chatbots have become increasingly popular, with a number of instant messaging services launching support for chatbots, making it very convenient to create chatbots for a number of applications. The developments in the field of NLP have lead to a growth in the number of intelligent tutoring systems that provide personalized learning environments to students.

Many studies [1] have proven that the use of chatbots for tutoring enhances student engagement in studies. Functionality to set and evaluate tests is an added benefit, as it enables teachers to seamlessly grade the performance of a large number of students.

The *Question Answering and Quiz Generation Chatbot* allows the user to upload relevant documents on which two main functions, namely answer extraction and question generation are performed, after converting the documents to a suitable knowledge base. The Question Answering module employs ranking functions to extract relevant answers from the knowledge base, out of which top K answers are further fed to a neural network which chooses the final answer. The Quiz Generation module uses a suite of ranking mechanisms to rank sentences based on their relevance to the subject matter in the document. The top K significant sentences chosen undergo NLP transformations and are then used to generate questions, which are further filtered and finally presented to the user as a quiz. The user responses to the questions are fed to the Answer Comparison module which calculates the score for the quiz. The proposed model is useful for subjects that contain fact-based knowledge (like Social Studies and Science) rather than subjects that require analytic or mathematical knowledge.

Most existing models face a number of challenges in designing a chatbot for Question Answering from a knowledge base supplied by the user, including lexical gap, context awareness and ambiguity. In a Question Answering system, a lexical gap is said to exist if the vocabulary used in a question posed by the user is different from the one used in the labels of the knowledge base. Context awareness refers to the ability of the system to retain the context of a conversation with the user. Ambiguity is the phenomenon in which the same word or phrase has different meanings depending on the context in which it is used. All these challenges have been addressed to an extent in our work.

The main advantage of using our chatbot over similar study aids is the flexibility provided to the user in terms of the documents that can be uploaded, and the provision to generate standardized quizzes to assess a large number of students.

This paper is organized into five sections as follows. Section 2 describes related work, Section 3 contains a detailed discus-

sion of our proposed model for the chatbot, Section 4 discusses the performance of our model along with comparisons against existing models, Section 5 summarises our work, and Section 6 concludes by discussing the scope for future work.

II. Related Work

A. Teaching using Technology

Percy [2] is a Teaching Assistant chatbot that answers questions posed by students on Piazza, an online forum for asking questions, by classifying questions as Policy, Assignment or Conceptual questions using linear SVM. AutoTutor [3] is a chatbot that imitates a human tutor by conversing with the student in natural language and helps the user frame better answers by prompting questions. CSIEC [4] is a computer-assisted English learning chatbot that helps users learn English by chatting in the English language using keywords or pattern matching to find the response for a given user text.

B. Question Answering

In [5], Apache PDFBox is used to extract text from PDF files using Optical Character Recognition (OCR). From the generated text document, question-answer pairs are generated using *Overgenerating Transformations and Ranking* algorithm. The user input is matched against the question-answer pairs using pattern matching to fetch the answer.

A feed-forward neural network is used in [6] to allocate scores to different passages according to the relevance of the passage to the question. InferSent (open source sentence embedding provided by Facebook Research for providing semantic representations of a sentence) representations are used to capture the general semantics of a question and the passage.

In [7], word features are used to generalize different retrieval tasks. The basic model uses TF-IDF vector space along with improvements such as low rank representations, correlated feature hashing, and sparsification.

C. Quiz Generation

Question Generation (QG) can be done using neural networks or NLP techniques. In our work, we consider the latter.

In [8] a method is proposed for generating questions from input text by overgeneration of questions from declarative sentences using syntactic transformations, and then ranking of these generated questions based on logistic regression model based ranker.

A study was conducted by [9], that identified Question Answering approaches, tools and systems, along with the metrics used to measure these approaches. A systematic literature review of work published from 2000 to 2017 indicated that precision and recall were the predominant evaluation metrics considered.

A comprehensive assessment of fifteen algorithms for sentence scoring relevant to text summarization was performed in [10]. The three main approaches for sentence scoring were word scoring, sentence scoring, and graph scoring.

III. Proposed System

The Question Answering and Quiz Generation Chatbot consists of two main modules - a Question Answering module and a Quiz Generation module. It has a functionality to upload textbooks, which undergoes a number of preprocessing steps and gets converted into a knowledge base that is used as input to these two modules. The document is converted from PDF to text format using Apache PDFBox, an open source Java tool. This text is then processed by removing irrelevant sentences using a rule-based approach and enforcing context awareness using Stanford NLPs CorefAnnotator [11].

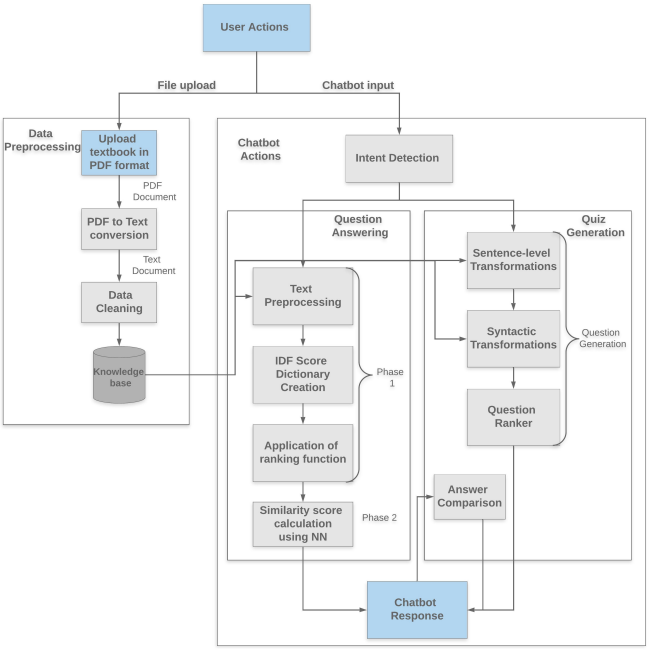

Fig. 1. The Proposed Model

A. Question Answering Module

The Question Answering module takes the user query as input and uses the knowledge base to extract the most appropriate answer sentence and returns it as a chatbot response. This is done in two phases.

1) Phase 1:

- *Text preprocessing:* Each sentence from the knowledge base and the user query goes through preprocessing steps including tokenization, singularization, lemmatization, and punctuation removal.
- *Inverse Document Frequency (IDF) Score Dictionary Generation:* Each sentence in the preprocessed text is taken as a document. The IDF score for each word is calculated as in Eq. 1.
- *BM25 Score calculation:* The user query is taken as the input and the top 10 answer sentences calculated using BM25 score (defined in Eq. 2 and 3) are returned.

$$idf[word] = log_e\left[\frac{N - freq[word] + 0.5}{freq[word] + 0.5}\right] \quad (1)$$

$$score+ = \left(\frac{idf[word] * (freq[word] * (k+1))}{freq[word] + k * (1 - b + b * \frac{sentenceLen}{avgSentenceLen})} \right) \tag{2}$$

$$termRank = c - d * log_e \left[\frac{(termPos - 1)/20 + 10}{sentenceLen/20 + 10} \right] \tag{3}$$

where idf[word] is the IDF score calculated; freq[word] is occurrence of the word in the answer sentence; sentenceLen is the number of words in the sentence; avgSentenceLen is calculated as above; k, b are parameters; termPos is the position of the word in the sentence; sentenceLength is the number of words in a sentence; c, d are parameters

2) Phase 2:

- The top 10 answer sentences are fed into a feed-forward neural network to pick the most relevant answer sentence.
- The query and the passages in the dataset are converted to their InferSent representations.
- The neural network is trained by minimizing the loss function given in Eq. 4.
- The weight matrix is updated as given in Eq. 5.
- The similarity score between the query and passage is found using the formula given in Eq. 6.

$$loss = \sum_{i=1}^{k} max(0, 1 - (q - a_{pos})^T W (q \odot a_{pos}) + (q - a_{neg})^T W (q \odot a_{neg})) \tag{4}$$

$$W \leftarrow W + \sum_{i=1}^{k} \lambda((q - a_{pos})(q \odot a_{pos})^T - (q - a_{neg})(q \odot a_{neg})^T) \tag{5}$$

where q is the question embedding, a_{pos} is the answer labelled correct, a_{neg} is the answer labelled wrong, W is the weight matrix, λ is the learning rate

$$score = (q - a)^T W (q \odot a) \tag{6}$$

where q is the question embedding, a is the answer embedding, W is the weight matrix

B. Quiz Generation Module

This module is responsible for (i) generating questions from an input document, and (ii) selecting a set of questions from those available, taking the user response, and calculating the quiz results after passing the responses through the answer comparison module.

1) Question Generation: This takes a document as input, converts it into a knowledge base and extracts top 100 sentences (chosen based on a suite of ranking functions), feeds them into a QG system, which returns a set of question-answer pairs such that each pair contains a chosen sentence as the answer and a Wh-question as its corresponding question. These pairs are further filtered before they are stored for retrieval for the quiz.

The first step in this process is the extraction of sentences. The type of questions considered in this work are those that fall into the bottom-most (sixth) level under the six levels of Blooms taxonomy (a model used to describe educational learning objectives and categorize them into levels of complexity) - remembering. Remembering refers to recalling information, and involves recognizing, listing, describing, retrieving, naming, and finding. Since our work only considers the "remembering" type of questions, the answer sentences to be chosen from the textbook should be fact-based, i.e., each question asked should have a valid fact as the main subject of the question. We identify key sentences and use only those to generate questions for the quiz. For this, a set of preprocessing steps coupled with a suite of scoring functions is employed. Sentences containing textbook related words such as "Chapter", "Section", "Activity", etc. are removed. This is done to eliminate the possibility of generating trivial questions like, *"What will you see in Fig. 1?"* , and is done as part of the general preprocessing step.

To identify the key sentences, the sentences in the textbook are ranked based on various parameters. Different types of scoring used are word scoring, which is based on word frequency, TF-IDF, uppercase, and proper noun; sentence scoring, which considers inclusion of numerical data, sentence length, and type of sentence (elimination of imperative sentences and rhetorical questions in source material); and graph-based scoring, namely TextRank. These scoring mechanisms are used as described in [10] with the exception of sentence length factor, which is used as described in Eq. 7.

$$\mathbf{LF(s)} = \frac{1}{|\textit{no. of words in s} - \textit{avg. no. of words in a sentence in d} + \text{c}|} \tag{7}$$

where s is the sentence being considered, d is the entire document under consideration, c is a small constant added to prevent a divide by zero error, and LF is the length factor score. The length factor scores of all the sentences are then normalised from 0 to 1.

After all these scores are computed for all the sentences in the document, the overall total score for each sentence is calculated as defined in Eq. 8.

$$totalScore = textRank^{e1} * typeOfSent * sentLen^{e2} * (a * wordFreq + b * tfIdf + c * upperCase + d * properNounWord + e * numericWord) \tag{8}$$

where textRank, typeOfSentence, sentLen, wordFreq, tfIdf, capWord, properNounWord, and numericWord stand for the scores from the individual scoring functions; e1 and e2 are exponents; and a, b, c, d and e are weightages to different scores.

The top 100 sentences ranked based on Eq. 8 are fed into the QG system, which is based on *Question Generation via Overgenerating Transformations and Ranking*, a technique outlined in [8]. It proposes a method to generate questions from declarative statements, and the process can be broken into stages.

In the first stage, a selected sentence is transformed into a declarative sentence through transformation of lexical items, syntactic structure, and semantics. Techniques such as extractive summarization, sentence splitting, fusion and compression, paraphrasing, and textual entailment may be used. In the second stage, the declarative sentence is converted to a question by syntactic transformations such as subject-auxiliary inversion, Wh-movement, etc. In the final stage, the questions are scored and ranked based on factors related to the source sentences, the question, and the transformations used in generation. Since many options are available at each stage of processing, this is an overgenerate-and-rank strategy. Since our work only considers fact-based questions, we restrict our system to generate only Wh questions.

The QG system generates a set of Wh questions for each sentence, such that either the sentence contains the answer, or the sentence itself is the answer. Out of this set of questions, one question is picked. After these question-answer pairs are returned by the QG system for each target sentence, they are filtered based on the length of the question, as compared to the length of the corresponding answer. If the length of the question is less than half of the length of the answer, there is a high chance that the question may not have enough information to answer. Hence, such pairs are further eliminated. This reduces the occurrence of vague questions such as, *What did Louis XVI do?*.

2) Quiz Generation: Quiz Generation refers to the access of the generated question-answer pairs and displaying them to the user. In advanced systems, if the questions are stored along with a rating indicating the difficulty level of the question, the quiz could be generated according to a certain level of difficulty.

3) Answer Comparison: For every question that is displayed by the chatbot, the user has to answer before proceeding to the next question in the quiz. When the user answers the question, the answer comparison module is responsible for calculating the score for that question. To look for an exact match between user and expected answers, cosine similarity may be used.

$$scoreDefault = cosineSimilarity(userAnswer, expectedAnswer) \quad (9)$$

However, this scoring mechanism will prove insufficient due to the problem of lexical gap. To account for lexical gap, the following steps are performed:

$$commonNounSet = set(\textit{nouns in expectedAnswer}) \bigcap set(\textit{nouns in question})$$

$$uncommonNounSet = set(\textit{nouns in expectedAnswer}) \setminus commonNounSet$$

$$scoreNoun = cosineSimilarity(\textit{nouns in userAnswer}, \textit{nouns in uncommonNounSet}) \quad (10)$$

Eq. 10 will give a more accurate score based on the amount of information the user has given in her answer. However, this would fail when the answer is a noun that is also present in the question. So, the final score is calculated as the maximum of both these scores.

$$scoreFinal = max(scoreDefault, scoreNoun) \quad (11)$$

IV. Performance Analysis

A. Question Answering

The word feature based model described in [7] which uses TF-IDF vector space model with a specific scoring function, and the InferSent ranker outlined in [6] which uses InferSent representations to capture general semantics of the query along with a scoring function followed by ReLU activation function are taken for comparison with our model. The InferSent ranker taken for comparison uses the aforementioned scoring function without the activation function.

All models are are trained with 2000 questions along with labelled answers, and tested with 500 questions from the MS MARCO dataset.

1) Evaluation metrics: Recall@k (k = 1, 3 and 5) and *Mean Reciprocal Rank* are the evaluation metrics used for comparison.

Recall@K is a measure of whether or not the correct answer is in the top K ranked answers.

$$Recall = \frac{TP}{TP + FN} \quad (12)$$

where TP (True Positive) is a measure of cases in which the correct answer is present in the top K ranked answers, FN (False Negative) is a measure of cases in which the correct answer is not present in the top K ranked answers.

Mean Reciprocal Rank (MRR) calculates the average of the reciprocal of the rank. MRR is calculated using the formula

$$MRR = \frac{1}{N} \sum_{i-1}^{N} \frac{1}{r_i} \quad (13)$$

where ri is the rank of the correct answer ranked by their scores, N is the number of questions tested.

TABLE I
Performance Analysis: Question Answering module

	Word feature based model	**InferSent ranker**	**Our approach (BM25+Infersent ranker)**
Recall@1	21.8	19.4	**23.8**
Recall@3	39.2	40.0	**51.8**
Recall@5	59.2	61.4	**70.2**
MRR	0.392	0.380	**0.435**

According to [12], queries can be of two types, namely precision-oriented and recall-oriented. It is stated that neural networks perform better on recall-oriented queries than on precision-oriented queries and the opposite is true for the classical BM25. Since our net score is the maximum of scores from both approaches, our system performs better than systems that use either of these two methods, as observed in Table 1.

B. Quiz Generation

Our work for Quiz Generation is compared with the baseline, which is the QG system proposed by [8]. While the baseline approach uses the first k sentences from the document as the input, our approach considers a preprocessed version of the document which is context aware, and applies a suite of ranking functions to pick the top K sentences to be used as input for the QG system.

QG systems are best evaluated by human annotation, as specified by [8]. Here, we have considered two metrics:

- *Answerability* of a question, a measure of how good the question is, in terms of information available in the question, as opposed to the information that should be present to answer the question
- *Suitability* of a sentence to be considered as a potential source text for a quiz question to aid in academic education

Both these metrics were defined and made available to 12 independent human annotators who rated the question-answer pairs output by the Quiz Generation system, which included 50 pairs generated from both the baseline approach and our approach. The different categories for distinguishing between the levels of answerability of a question was considered as defined in [13] and shown in Table 2. We defined a similar metric called suitability, and fixed specific categories based on which the text under consideration could be determined for relevance for a quiz, as shown in Table 3. Each level or rating comprises of a specific description along with an example based on Class IX CBSE history textbook.

TABLE II
DETERMINING THE ANSWERABILITY OF A QUESTION

Rating	Description	Example
1	Cannot answer the question, absolutely no information available	*What did not all citizens have?*
2	Cannot answer the question, there is some information, but not sufficient	*Who ordered troops to move into Paris at the same time?*
3	Ambiguous answers possible due to missing information	*When did the situation in France continue to be tense?*
4	Can answer the question based on information present in the question	*What was a way of proclaiming the end of the power wielded by the wearers of knee breeches?*
5	Can answer the question, all information is present	*When did the Jacobins plan an insurrection of a large number of Parisians who were angered by the short supplies and high prices of food?*

The results from Fig.2 show that the procedure of enforcing context awareness and filtering out trivial sentences from the content leads to an improvement in the average answerability of questions generated.

The results from Fig. 3 show that the procedure of ranking sentences leads to an improvement in the average suitability of the text for a quiz.

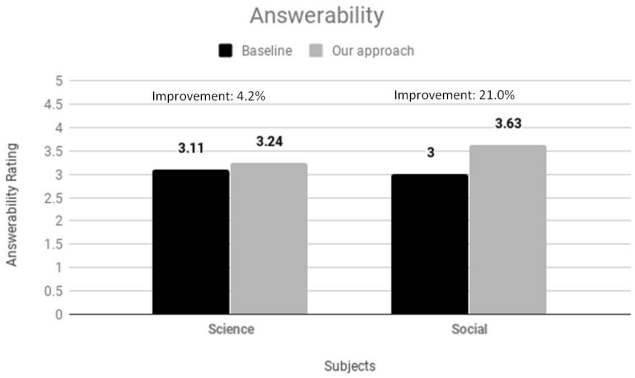

Fig. 2. Answerability Rating Graph

TABLE III
DETERMINING THE SUITABILITY OF A SENTENCE FOR A QUIZ

Rating	Description	Example
1	Text does not contain information that could be used to test the knowledge of a student in the subject or text has a lot of missing information that a meaningful question (relevant to the subject) cannot be formed from it	*They all described and discussed the events and changes taking place in France.*
2	Text contains a lot of missing information, and may not be able to test the knowledge of the student in the subject, even if the missing information is made available	*At the same time, the king ordered troops to move into Paris.*
3	Text conveys useful information, but there is missing information leading to difficulty in forming questions out of it, but may be able to test the knowledge of a student, to some extent in the subject, if missing information is made available	*Lenders who gave the state credit, now began to charge 10 per cent interest on loans.*
4	Text conveys information that can be considered important, has no missing information, and can test the knowledge of a student of the intended grade, to some extent in the subject	*Under Louis XVI, France helped the thirteen American colonies to gain their independence from the common enemy, Britain.*
5	Text conveys a crucial piece of information, has no missing information, and can test the knowledge of a student of the intended grade, significantly in the subject	*In his Two Treatises of Government, Locke sought to refute the doctrine of the divine and absolute right of the monarch.*

Hence it is observed that both the Question Answering and Quiz Generation modules show an improvement over the baseline models considered.

V. CONCLUSION

Our work was aimed at developing a study aid in the form of a chatbot that can assist primary and middle school students in learning fact-based subjects. The chatbot can take an uploaded document as input and enable the student to ask questions from the text or request for quizzes to test her knowledge.

The quality of Question Answering was enhanced by giving weightage to both the semantic properties as well as the word

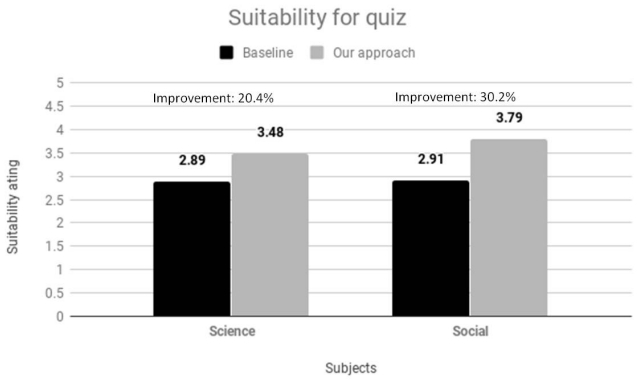

Fig. 3. Suitability Rating Graph

relevance of the sentence to the question posed, considering scores obtained through BM25 as well as the feed-forward neural network. Thus, both precision-oriented and recall-oriented queries are handled well by the chatbot.

The quality of Quiz Generation was enhanced by identifying sentences that are likely to give rise to good quiz questions. This was done by eliminating non-declarative sentences, enforcing context awareness within the content in the document, and applying a suite of scoring mechanisms to rank sentences that finally yielded a set of question-answer pairs highly relevant to the subject matter.

VI. Future work

The following can be done to extend our work.

- *Scrapping relevant websites* for additional information can supplement the existing knowledge base
- *Feedback-based improvement* can be incorporated in quizzes for the user to learn at her own pace
- An *encouraging personality* introduced in the chatbot can motivate the user
- *Consolidation of relevant data* from all the uploaded documents may enable the system to return answers to complex questions

References

[1] Roos, S. (2018). Chatbots in education: A passing trend or a valuable pedagogical tool?.

[2] Chopra, S., Gianforte, R., Sholar, J. (2016). Meet Percy: CS 221 Teaching Assistant Chatbot. ACM Transactions on Graphics, 1(1).

[3] Graesser, A. C., Chipman, P., Haynes, B. C., Olney, A. (2005). AutoTutor: An intelligent tutoring system with mixed-initiative dialogue. IEEE Transactions on Education, 48(4), 612-618.

[4] Jia, Jiyou. (2009). CSIEC: A computer assisted English learning chatbot based on textual knowledge and reasoning. Knowledge-Based Systems. 22. 249-255. 10.1016/j.knosys.2008.09.001.

[5] Pichponreay, L., Choi, C., Kim, J., Lee, K., Cho, W. (2016). Smart answering Chatbot based on OCR and Overgenerating Transformations and Ranking. 2016 Eighth International Conference on Ubiquitous and Future Networks (ICUFN), 1002-1005.

[6] Htut, P.M., Bowman, S.R., Cho, K. (2018). Training a Ranking Function for Open-Domain Question Answering. NAACL-HLT.

[7] Bai, B., Weston, J., Grangier, D. et al. Learning to rank with (a lot of) word features. Inf Retrieval (2010) 13: 291.

[8] Heilman, M., Smith, N. A. (2010, June). Good question! Statistical ranking for question generation. In Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics (pp. 609-617). Association for Computational Linguistics.

[9] Calijorne Soares, M. and Parreiras, F. (2018). A literature review on question answering techniques, paradigms and systems. Journal of King Saud University - Computer and Information Sciences.

[10] Ferreira, R., de Souza Cabral, L., Lins, R. D., e Silva, G. P., Freitas, F., Cavalcanti, G. D., ... Favaro, L. (2013). Assessing sentence scoring techniques for extractive text summarization. Expert systems with applications, 40(14), 5755-5764.

[11] Recasens, M., de Marneffe, M. C., Potts, C. (2013, June). The life and death of discourse entities: Identifying singleton mentions. In Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (pp. 627-633).

[12] Nakamura, T. A., Calais, P. H., de Castro Reis, D., Lemos, A. P. (2019). An anatomy for neural search engines. Information Sciences, 480, 339-353.

[13] Nema, P., Khapra, M. M. (2018). Towards a Better Metric for Evaluating Question Generation Systems. arXiv preprint arXiv:1808.10192.

[14] Madrid, N. and Rusnok, P. (2019). A Top-K Retrieval algorithm based on a decomposition of ranking functions. Information Sciences, 474, pp.136-153.

[15] Sharma, Y., Gupta, S. (2018). Deep Learning Approaches for Question Answering System. Procedia computer science, 132, 785-794.

[16] Song, L., Wang, Z., Hamza, W. (2017). A unified query-based generative model for question generation and question answering. arXiv preprint arXiv:1709.01058.

[17] Duan, N., Tang, D., Chen, P., Zhou, M. (2017, September). Question generation for question answering. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (pp. 866-874).

[18] Tang, D., Duan, N., Qin, T., Zhou, M. (2017). Question Answering and Question Generation as Dual Tasks. CoRR, abs/1706.02027.

[19] Xing, C., Wu, W., Wu, Y., Liu, J., Huang, Y., Zhou, M., Ma, W. Y. (2017, February). Topic aware neural response generation. In Thirty-First AAAI Conference on Artificial Intelligence.

[20] Xing, C., Wu, W., Wu, Y., Liu, J., Huang, Y., Zhou, M., Ma, W. Y. (2017, February). Topic aware neural response generation. In Thirty-First AAAI Conference on Artificial Intelligence.

[21] Nguyen, T., Rosenberg, M., Song, X., Gao, J., Tiwary, S., Majumder, R., Deng, L. (2016). MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. CoRR, abs/1611.09268.

[22] Zhou, H., Huang, M. (2016, December). Context-aware natural language generation for spoken dialogue systems. In Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers (pp. 2032-2041).

[23] Trotman, A., Puurula, A., Burgess, B. (2014). Improvements to BM25 and Language Models Examined. ADCS.

[24] Liu, M., Calvo, R. A., Rus, V. (2010, June). Automatic question generation for literature review writing support. In International Conference on Intelligent Tutoring Systems (pp. 45-54). Springer, Berlin, Heidelberg.

[25] Wang, B., Wang, X., Sun, C., Liu, B., Sun, L. (2010, July). Modeling semantic relevance for question-answer pairs in web social communities. In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (pp. 1230-1238). Association for Computational Linguistics.

[26] Robertson, Stephen Zaragoza, Hugo. (2009). The Probabilistic Relevance Framework: BM25 and Beyond. Foundations and Trends in Information Retrieval. 3. 333-389. 10.1561/1500000019.

[27] Svore, K.M., Burges, C.J. (2009). A machine learning approach for improved BM25 retrieval. CIKM.

[28] Prez-Iglesias, J., Prez-Agera, J.R., Fresno-Fernndez, V., Feinstein, Y.Z. (2009). Integrating the Probabilistic Models BM25/BM25F into Lucene. CoRR, abs/0911.5046.

[29] Cong, G., Wang, L., Lin, C., Song, Y., Sun, Y. (2008). Finding question-answer pairs from online forums. SIGIR.