Journal of the ACS Advances in Computer Science
Volume 16, Issue 1, 2025

A Systematic Review of Automatic Neural Question Generation

Asmaa M. Abdelwahab, Mahmoud M. Eid
Higher Institute of Computers and Information Technology, Computer Science Department, El-Shorouk Academy, Cairo, Egypt
Email: asmaa.abdelwahab@sha.edu.eg, mahmoud.eid@sha.edu.eg,

[1] ***Abstract***
The ability to formulate meaningful questions is a fundamental aspect of both human and artificial intelligence. Neural Question Generation (NQG) uses deep learning techniques to automatically generate relevant questions from a given context. NQG systems have significant applications in improving question-answering models, facilitating educational tools, and enhancing conversational agents such as chatbots. However, a key challenge in NQG is the effective selection of target sentences and concepts for question formulation. This paper presents a systematic literature review (SLR) of NQG, analyzing different datasets, input preprocessing methods, methodologies, and evaluation techniques. We also highlight emerging trends and future directions in the field. Our review provides a comprehensive overview of NQG research, offering insights into current progress and remaining challenges. We find that all NQG models share a common Seq2Seq framework. In addition, the integration of Seq2Seq with attention mechanisms, as well as the use of part-of-speech (POS) tagging and named entity recognition (NER), contributes to the generation of accurate questions.

Index Terms— Natural Language Processing (NLP), Neural Question Generation (NQG), Deep Neural Networks, Question Answering Systems, Systematic Literature Review (SLR).

1. INTRODUCTION

Natural Language Processing (NLP) is a central subfield of computer science and artificial intelligence that focuses on enabling computers to understand and interact with human language. A fundamental challenge in NLP is training machines to process and analyze large amounts of natural language data (Joseph, 2016) (Sarkar, S,2025).The overall goal is to develop systems that can understand the content of various text formats, including sentences, queries, paragraphs, and documents. NLP techniques facilitate tasks such as text classification, where textual units are assigned labels or tags (Yang, 2020) (Maity, 2025). Applications of NLP range from answering questions and spam detection to sentiment analysis and news categorization. The sources of text data are diverse, including web content, email, forums, social media, and user reviews.

Automated text classification uses a variety of methods, including rule-based techniques and machine learning algorithms such as decision trees, naive Bayes, and k-means clustering (Semerikov, S. O,2025) (Shervin et al., 2021). In addition, deep learning approaches, particularly those using neural networks, have gained prominence. Preprocessing steps, which can include punctuation removal, word segmentation, stop word filtering, and stemming (Elbes, 2019) (AlKhuzaey,2024), are critical to improving classifier performance. Feature selection methods, including information gain (IG), expected cross entropy (ECE), mutual information (MI), Gini index (GI), and chi-square (CHI) (Mucciaccia,2025) (Bennabi, 2020), also play an important role in optimizing results.

This paper primarily focuses on neural Question Generation (QG), which uses deep neural networks to automatically generate questions from various inputs, such as raw text, databases, or semantic representations. Historically, question generation has relied on heuristic methods that rely heavily on human-designed transformation and generation rules, making it difficult to adapt to different domains (Heilman, 2011; Chali and Hasan, 2015). In contrast, Serban et al. (2016) introduced a neural network approach for generating factual questions from structured data. The ability to generate effective questions is crucial for assessing knowledge and promoting self-directed learning in educational contexts. In addition, QG can improve question-answering systems and enable chatbots to engage more dynamically in conversations.

The rest of this paper is organized as follows: The second section outlines the research methodology, specifically a Systematic Literature Review (SLR). The third section covers input text preprocessing techniques, including tokenization, segmentation, and the use of NLP tools such as word embeddings, Part-Of-Speech (POS) tagging, and Named Entity Recognition (NER). We will also explore neural question generation models, such as the sequence-to-sequence (seq2seq) model using Gated Recurrent Units (GRU) and seq2seq models with attention mechanisms. In addition, we will discuss evaluation metrics, including BLEU and precision. Finally, the fourth section presents conclusions and suggestions for future research.

2. METHODOLOGY

The research approach follows the Systematic Literature Review (SLR) guidelines for the discipline of computer engineering as proposed by Kitchenham and Charter (Kitchenham, 2012). Figure 2.1 illustrates the key stages of our process, which we will discuss in the following sections.

In the planning stage, we defined our research topics and established the basic elements of the review protocol. To minimize subjectivity, we required that each phase begin only after the previous one had been thoroughly evaluated and approved.

The search strategy includes the criteria for selecting studies, the methods used, the search strings used, and the assessment of study quality. A significant portion of the third phase is devoted to developing our data extraction strategy. Finally, the final phase of the systematic review involves the preparation of a synthesis matrix to summarize and analyze the results.

Figure 2.1: The key stages of our process (SLR).

2.1 Research Questions

2.1.1 Old Research Questions:

[1] What is text classification and when did it originate?
[2] What are the different applications of text classification?
[3] Which languages are of interest for classification in this research?

[4] What are the limitations and challenges of existing text classification methods?
4.1 What are the limitations and challenges of existing automatic text classification (ATC) methods?
[5] What data sets are available to evaluate model performance?
5.1 What is an appropriate approach to document representation?
5.2 What are the different types of data preprocessing techniques?
5.3 How can unstructured data be handled effectively?
[6] What are the appropriate methods for feature extraction and feature selection?
[7] What techniques are used for text classification, especially for news categorization and question answering applications? Which models perform best in these areas?
7.1 What techniques are used in ATC and which models are most effective?
[8] What solutions exist to improve the performance of current techniques?
[9] What methods are used to evaluate the performance of text classification models?
[10] What are the future research directions in text classification, especially in automatic text classification (ATC)?

2.1.2 New Research Questions:

The primary objective of this research will be achieved by answering the following questions:

RQ1. What is Neural Question Generation (NQG)?
RQ2. What are the different applications of NQG?
RQ3. Which languages are of interest for NQG research?
RQ4. What are the limitations and challenges of existing NQG methods?
RQ5. What benchmark data sets are available to evaluate the performance of NQG models?
RQ5.1 What is an appropriate approach for representing words?
RQ5.2 What are the different types of input preprocessing techniques?
RQ6. Which techniques are used in NQG?
RQ7. What are some possible solutions and strategies to improve the performance of current NQG methods?
RQ8. What methods are used to evaluate the performance of NQG models?
RQ9. What are future research directions for NQG?

2.2 Data Sources and Search Strategy

Table 2.1 illustrates how we identified research publications in computer science and software engineering using various database sources. The search terms were defined to include the following keywords, which were generated using logical operators to optimize the search results:

1. Search string 1: ('document' OR 'text') AND ('classifier' OR 'categorization' OR 'classification') AND ('document representation' OR 'document preprocessing' OR 'models' OR 'methods' OR 'application' OR 'evaluation' OR 'assessment' OR 'challenges' OR 'limitations').
2. Search string 2: ('document' OR 'text') AND ('classifier' OR 'categorization' OR 'classification') AND ('research') AND ('future' OR 'trend' OR 'direction').
3. Search string 3: ('characteristic' OR 'attribute') AND ('selection' OR 'selected') AND ('text' OR 'document') AND ('classification' OR 'classifier' OR 'categorization').

These search strings were adapted to take advantage of the built-in tools for refining and filtering results in each database. In addition, we included gray literature and used a snowballing approach where each publication identified by our search criteria could be manually linked to other relevant citations in its references.

Database	URL
ACM	ACM Digital Library
IEEE	https://ieeexplore.ieee.org/
Springer	http://link.springer.com/
Semantic Scholar	https://www.semanticscholar.org/

Table 2.1: Databases

2.3 Study Selection Criteria

To ensure the collection of quality and relevant data in response to our research questions, we implemented strict inclusion and exclusion criteria during this step. These criteria were applied after reviewing the title, abstract, and full text of each article.

Inclusion Criteria:

- The paper is relevant to text classification (e.g., news categorization, question answering).
- The publication date is between 2016 and 2024.
- The paper highlights one or more problems, weaknesses, or limitations of existing text classification techniques, along with proposed solutions.
- It uses relevant keywords.
- It is related to the Arabic and/or English languages.
- The paper is written in English.

Exclusion criteria:

- Articles not written in English.
- Articles published before 2016 or after 2024.
- Text classification models that are not evaluated using Arabic or English.

2.4 Study Selection Process

The primary study selection process consisted of three separate steps, as shown in Figure 2.2. Applying the search string to the four scientific databases listed in Table 2.1 generated over 2,000 articles.

- **Iteration 0 - Filtering by Title**

In this phase, the titles were evaluated against the inclusion and exclusion criteria. Articles deemed relevant were immediately included in the next phase. A total of 230 publications were selected for further review.

- **Iteration 1 - Filtering by Abstract and Keywords**

In this phase, the abstracts and keywords were evaluated against the inclusion and exclusion criteria. Articles considered relevant were included in the next stage. A total of 220 publications were selected in this phase.

- **Iteration 2 - Filtering by Full Text**

This was the final step in which the full texts were examined based on the quality assessment criteria (as detailed in section E). We ranked the papers from the previous step and selected the top 15 for further review. My supervisor then provided me with 6 key papers. Finally, 21 articles were included in the final step.

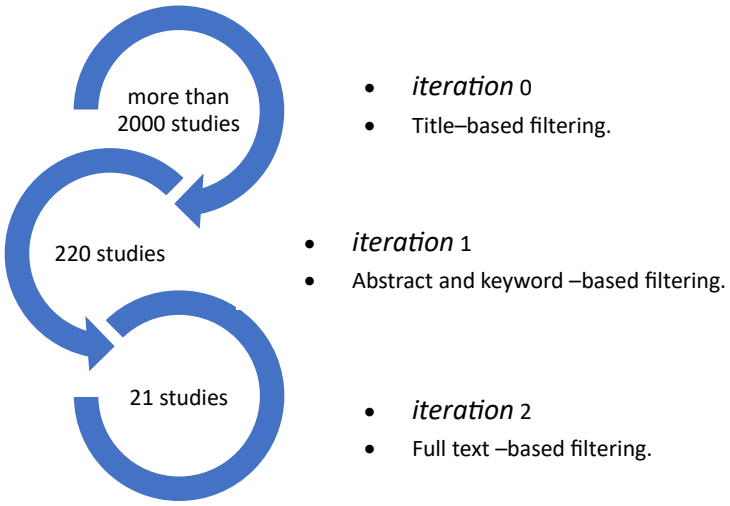


Figure 2.2: The number of studies included in each study selection phase.

2.5 Quality Assessment

Included papers had to pass a quality check, which included positive answers to the following questions

- What is the impact factor of the journal or conference?
- What is the number of citations?
- Is the data set clearly identified and well described?
- Are the preprocessing techniques used in the study well described and their selection justified?
- Is the total number of training and test data provided?
- Are the classifiers used in the study discussed in detail?
- Is there a comparison of different approaches?
- Are performance metrics defined in detail?

Ref	QA1	QA2	QA3	QA4	QA5	QA6	QA7	QA8	Total	percentage
(Minaee,2021)	1	1	1	1	0.5	1	1	1	7.5	93.75%
(Alabbas,2016)	1	0.5	1	1	0.5	1	1	1	7	87.50%
(Ahlam Wahdan,2020)	1	0.5	1	1	0.5	1	1	1	7	87.50%

(Abdeen,2019)	1	0.5	1	1	0.5	1	1	1	7	87.50%
(Bennabi,2020)	1	0.5	1	1	0.5	1	1	1	7	87.50%
(Al Qadi, Leen,2019)	1	0.5	1	1	0.5	1	1	1	7	87.50%
(Ezzeldin,2012)	1	1	1	1	0	1	1	1	7	87.50%
(Zhang,2021)	1	0.5	1	1	0.5	1	1	1	7	87.50%
(Elbes,2019)	1	0.5	1	1	0.5	1	1	1	7	87.50%
(Rachid,2020)	1	0.5	1	1	0.5	1	1	1	7	87.50%
(Pandolfi,2020)	1	0.5	1	1	0.5	1	1	1	7	87.50%
(Wahdan,2021)	1	0.5	1	1	0.5	1	1	1	7	87.50%
(Liu,2020)	1	0.5	1	1	0.5	1	1	1	7	87.50%
(Kadhim,2019)	1	1	0.5	1	0	1	1	1	6.5	81.25%
(Shehab,2016)	1	0.5	1	1	0.5	1	0.5	1	6.5	81.25%

Table 2.2: QA Paper

2.6 Included papers

In this section, we present the distribution of included papers based on the following criteria: database, year, type (journal or conference), and publisher.

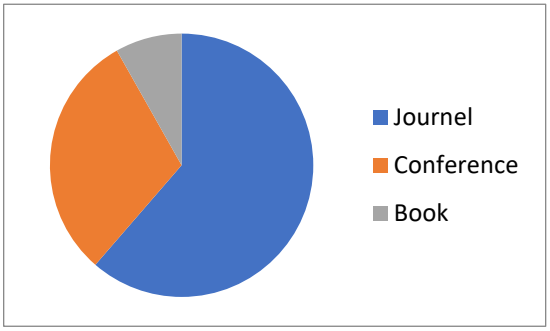


Figure 2.1: distribution based on type

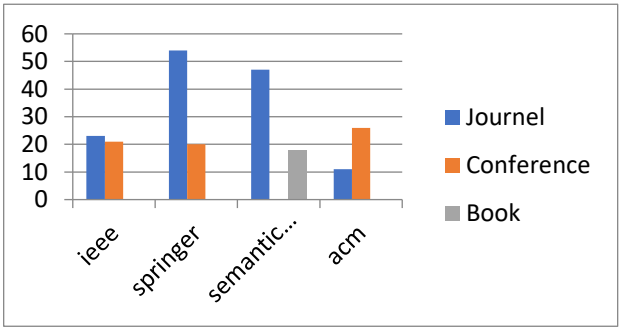


Figure 2.2: based on databases

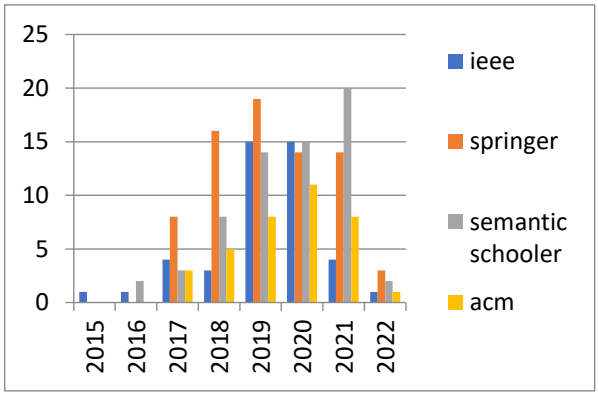


Figure 2.3: distribution based on year

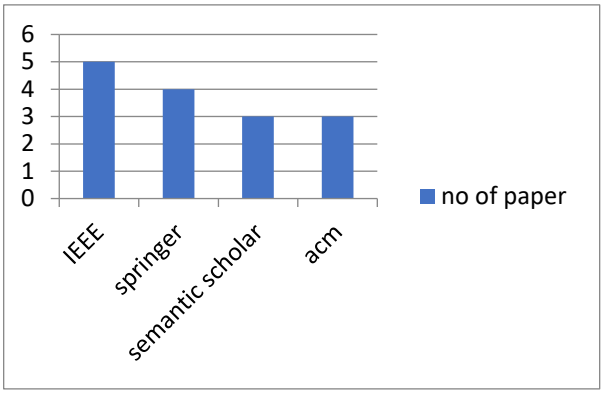


Figure 2.4: Distribution of top 15 papers

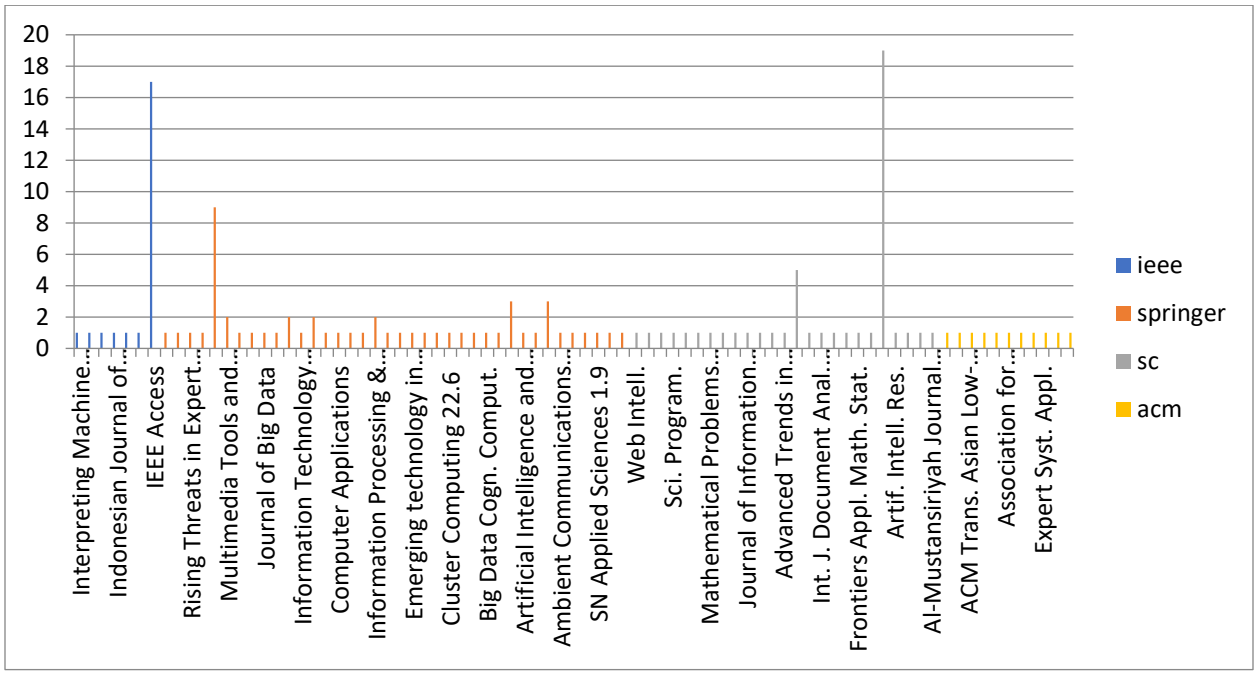


Figure 2.5: distribution based on publisher

2.7 Data Extraction Strategy

Data were extracted from the studies, and Table 2.3 shows the characteristics that were collected and included:

Paper Information:

- Title Author
- Publication year
- Journal name
- Study type

Data extracted:

The form included the following characteristics:

- Paper Objective
- Excerpts from Introduction
- Excerpts from Literature Review
- Question Generation (QG) Algorithm
- Evaluation Metrics
- Conclusion
- Research Questions Addressed

The data extracted from the sample papers is shown in Table 2.4.

Paper Information	
1. Title	
2. Author	
3. Year	
4. Journal	
5. Study Type	
Extracted Data	
6. Which research question will be answered?	
7. Objective extracted from the study	
8. Extracted pieces from Introduction	
9. Extracted pieces from Literature review	
10. Extracted pieces from EXPERIMENTAL STUDY (optional)	
11. Conclusion Extracted	

Table 2.3: DE Form

Paper Information	
1. Title	Learning to Ask: Neural Question Generation for Reading Comprehension
2. author	Du & Shao (2017)
3. Year	*2017*
4. journal	*arXiv preprint arXiv:1705.00106* (2017).
5. Study Type	Experimental Study
Extracted Data	
6. Which research question will be answered?	RQ1 – RQ2 – RQ4 – RQ6– RQ7 - RQ8
7. Objective extracted from the study	In reading comprehension, investigate automatic question generation for sentences from text passages.

8. Extracted pieces from Introduction	• Question generation (QG) is a technique for generating natural questions from a sentence or paragraph. • One of the most common uses of question creation is in the field of education, where it is used to produce reading comprehension questions. • Question generation has traditionally been approached using rule-based methodologies. • In contrast to previous work, we propose framing the task of question production as a sequence-to-sequence learning issue, in which a sentence from a text passage is immediately translated into a question.
9. Extracted pieces from Literature review	• Reading comprehension is a difficult challenge for robots since it necessitates both a grasp of natural language and a knowledge of the world. • The majority of work takes a rule-based approach to solving the problem. • They usually start by converting the input sentence to its syntactic representation, which they then utilise to create an interrogative sentence. • To our knowledge, no previous work has employed a deep sequence-to-sequence learning approach to generate questions or framed QG for reading comprehension in an seq-to-seq manner.
10. Extracted pieces from EXPERIMENTAL STUDY and Results	<u>Dataset :</u> • SQuAD dataset <u>Perprocessing :</u> • first use Stanford CoreNLP Tokenization and sentence splitting are used for pre-processing. The full dataset is then lower-cased. <u>Model :</u> • encoder :encode both sentence and paragraph-level information with attention mechanism. • Decoder : Decodes the questions using the concatenated representation. <u>Evaluation Metrics :</u> • Automatic Evaluation • Human Evaluation <u>Results:</u> • The proposed model, which simply encodes sentence-level information, outperforms all others on all criteria. • The proposed model, which encodes paragraph information, performs best on questions in the "w/ paragraph" category.

11. Conclusion	• It demonstrated a technique to autonomous question development for reading comprehension based entirely on data-driven neural networks. • Using an attention-based neural networks method, we investigate the effect of encoding sentence-level vs. paragraph-level information. • The suggested model produces state-of-the-art results in both automatic and human evaluations.

Table 2.4: Data extracted from sample paper

2.8 Synthesis Matrix

The synthesis matrix for the studies is tabulated as shown in Table 2.5. It includes the following characteristics:

- Author
- Publication Year
- Paper Objective
- Preprocessing Methods (if applicable)
- Datasets Used
- Question Generation (QG) Algorithms/Methodology
- Evaluation Metrics
- Conclusion

Source *(Author//Year)*	Application Type(QA/QGP/QG /NC/Survey)	Purpose of study	Preprocessing Methods	Datasets	Methodology/Algorithm	Results/ Evaluation Metrics	Conclusion
Derwin Suhartono,2024	automatic question generation	to compare several state-of-the-art pre-trained models to create an automatic question generator with narrative paragraphs as input.	-the paragraph is broken down into smaller units called tokens. -These tokens can be individual words, punctuation marks, or other meaningful units -Each token is assigned a part-of-speech tag, specific answers are extracted	SQuAD, TyDiQA, IDK-MRC Datasets	uses the Sequence-to-Sequence Learning architecture of BiGRU, BiLSTM, Transformer, BERT, BART, and GPT	BLEU-1 : IndoBERTFormer 29.14 IndoBERTFormer 30.45	- this research only evaluates these models for the case of creating short answer questions this research only uses three trained models: Indo BERTFormer, IndoBARTFomer, IndoTransGPT, Our methodology efficiently uses context-to-

			from the postage tensor based on specific criteria or questions.				answer attention more reliably than longer answers to extract more relevant information from surrounding sentences
Xinya Du,2017	Question Generation	In reading comprehension, investigate automatic question generation for sentences from text passages.	To begin, perform pre-processing with Stanford CoreNLP, which includes tokenization and sentence splitting. The full dataset is then lower-cased.	SQuAD dataset	encoder: use an attention technique to encode information at the sentence and paragraph levels. Decoder: decodes the questions using the concatenated representation.	Metrics : AE,HE Results: The proposed approach obtains the greatest results by just encoding sentence-level information.	In both automatic and human evaluations, the proposed model (encoder – decoder with attention mechanism) achieves state-of-the-art performance.
Wang,]2020	Question Generation	Based on an encoder-decoder framework and reinforcement learning, we propose an ADDQG model. Questions can be generated from answers and document representations.	Use pre-trained GloVe word vectors for word embedding.	HotpotQA,	The model's main idea is to merge the answer information with the content using an answer-aware initialization module and a semantic rich fusion attention mechanism. In addition, reinforcement learning is used. Using the Maxout Pointer and the Copy Mechanism	Metrics Automatic Evaluation: BLEU, METEOR, ROUGE Human Evaluation	Reinforcement learning is also used to improve ADDQG training by using both syntactic and semantic metrics as the reward.
Zhou,2017	Question Generation	It proposes that natural language sentences be used to generate relevant and diversified queries using the neural encoder decoder architecture.	--	SQuAD dataset	The NQG framework consists of a feature-rich encoder and an attention-based decoder. The BiGRU encoder reads the concatenated sentence word vector, lexical characteristics, and answer position feature. Copy Mechanism	This demonstrates how lexical features and an indicator of answer position can help with question development. With the assistance of the copy mechanism,	The suggested technique uses a feature-rich encoder to encode answer location, POS, and NER tag information. Experiments demonstrate that the proposed NQG technique is effective.
Yao, 2021	Question and Answer Pair Generation	Create an educational automated	--	Using FairytaleQA, a new QA	Methodology: The QAG system in this paper consists	Results All of the data reveal that our	The work sets a strong foundation for

		question-answer generation (QAG) system. The technology can generate QA pairings that can be used to assess a range of student comprehension skills automatically.		dataset with 278 kid-friendly storybooks and 10,580 expert-labeled QA pairs.	of three steps: (1) extracting candidate answers from given storybook passages using carefully designed heuristics based on a pedagogical framework; (2) generating appropriate questions corresponding to each of the extracted; and (3) ranking top QA-pairs with a specific threshold for the maximum amount of QA-pairs for each section..	approach receives above-average 601 (>3) ratings, implying that it achieves acceptable levels of user satisfaction across all three aspects (Readability, Question Relevancy , Answer Relevancy).	the bright future of applying artificial intelligence to automate educational question-answering chores.
Pan, 2019	Survey(NQG)	give a thorough examination of the corpora, methodology, and evaluation methods for neural question generation	--	SQuAD, MS MARCO,NewsQA,RACE,LearningQ,NarrativeQA.	give a thorough examination of the corpora, methodology, and evaluation methods for neural question generation In passage X, asking about the goal response A is defined as finding the optimal question Y. The Seq2Seq framework is shared by all NQG models, but they differ in how they consider (1) QG-specific characteristics (for example, response encoding, question word formation, and paragraph-level contexts) and (2) common NLG techniques (e.g., copying mechanism, linguistic features, and reinforcement learning).	Human evaluation is used in the majority of QG systems. BLEU, METEOR, and ROUGE are examples of automatic evaluation metrics.	This research offered a comprehensive overview of NQG, identifying current NQG models based on QG-specific and common technical changes, and highlighting three growing NQG trends: multitasking, a wider range of input modalities, and the development of profound questions.. In many real-world applications, such as automated tutoring and conversational systems, where the question plays a crucial role, knowing when to enquire has become a vital difficulty.

Benaissa Azzeddine Rachid,2020	News Classification	Researchers used neural network models (Convolutional and Recurrent Neural Networks) and pre-trained word embeddings in a series of experiments to classify cyberbullying situations using an Arabic channel news comments dataset.	Punctuation in Arabic and English has been removed. word embeddings as a source of data for deep learning algorithms	Aljazeera.net, an Arabic news station.	CNN, LSTM GRU combination of both. SVM	The findings show that using simple and combined Convolutional and Recurrent Neural Networks (CNN/LSTM/ GRU) with Arabic pre-trained word embeddings (AraVec and Fast text) combined with Arabic pre-trained word embeddings (AraVec and Fast text) combined with Arabic pre-trained word embeddings (AraVec and Fast text) can achieve an F1 score of 84 percent on a balanced dataset..	techniques of CNN-RNN , both simple and mixed, perform well.
Menghan Zhang,2021	News Classification	the concept of a customised algorithm, which is a mix of deep learning algorithms such as CNN and LSTM	Using the word2vec model, word segmentation and stop word filtering	Reuters News	For the classification of news text data, a bespoke DCLSTM-MLP model was used.	Accuracy of DCLSTM-MLP is 94%	The DCLSTM-MLP model outperforms the CNN and LSTM models in terms of accuracy.
Ahmed Magdy ,2012	Answer Generation(survey)	--	Stemming, Named Entity Recognition	SQuAD	--	--	--
Leen Al Qadi, 2019	News Articles classification	To automatically determine a document's category.	--	---	Famous techniques in classification were used: Logistic Regression, Nearest Centroid, Decision Tree (DT), Support Vector Machines (SVM), K-nearest neighbors (KNN), XGBoost Classifier, RandomForest Classifier, Multinomial Classifier, Ada-Boost Classifier,	--	Among all the other classifiers, the SVM model generated the best results.

					and Multi-Layer Perceptron (MLP).		
Shervin Minaee,2021	TC(A Comprehensive Review)	provide a quantitative analysis of various deep learning models' performance on popular benchmark datasets	--	Sentiment Analysis (Yelp,IMDb) NCDatasets(AG News, 20 Newsgroups.	Naïve Bayes, (SVM), hidden Markov model (HMM), gradient boosting trees, and random forests	--	--
Sakina Rim BENNABI,2020	FS(Comparative Study)	The goal of this paper is to give a comparison of several feature selection strategies.	--	--	Classification algorithms: SVM, KNN and NB.	--	--
Mohammad A.R. Abdeen, 2019	ATC (Review Paper)	a thorough examination of the Arabic text classification: The methodology, datasets, and feature selection strategies described in this paper	Normalization stemming algorithms	--	TC Methods : Decision Trees: Naive Bayesian k-means algorithms Hierarchical clustering algorithms.(better than K-means).	--	--
Ahlam Wahdan,2020	ATC(Systematic Literature Review)	examining neural network-based Arabic text categorization	--	--	Classification Techniques : Techniques that are both manual and statistical. Machine learning techniques	--	--
W.Alabbas, 2016	ATC(Systematic Literature Review)	Arabic text is classified using a variety of TC approaches and methods.	--	--	SVM,NB, Decision-tree, k-NN	--	--
Ammar Ismael Kadhim,2019	Survey(ML for TC)	Text classification surveys, the process of varying term weighing strategies, and a comparison of alternative categorization procedures.	--	--	Naïve Bayes, SVM KNN	--	--
Shehab,2016	multilabel classification of Arabic articles	focuses on Arabic article multilabel categorization	--	--	classifiers are considered (DT, RF and KNN).	--	--
Ahlam Wahdan, 2021	ATC	The goal of this study is to see how deep learning affects ANLP text classification.	--	--	Many techniques, such as word embedding and deep learning, have been employed to improve the application of	--	--

					natural language processing.		
Mauricio, 2021	Web News	News-related data was gathered from the web and classified using machine learning and data mining techniques.	Stop words removal, Stemming, Tokenizing	--	Clustering, support vector machines, and generative models were the three most common paradigms discovered.	--	--
Mohammed Elbes,2019	P-Stemmer or NLTK Stemmer for Arabic Text Classification ?	Using the above-mentioned categorization technique, we compared the outcomes of two stemmers: P-Stemmer and NLTK stemmer.	Preprocessing : P-Stemmer and the NLTK	--	--	--	--
Liu,2020	Comparison on Feature Selection Methods for Text Classification	Discussions to compare the performance of common feature selection strategies used in text classification studies in the past.	Feature selection techniques : Information gain (IG) Expected cross entropy (ECE) mutual information (MI) Gini index (GI) The core of Chi-square (CHI) The core of Odd ration (OR)	--	--	--	--

Table 2.5: Synthesis Matrix

3. RESULT AND DISCUSSION

3.1 Neural Question Generation (NQG)

The NQG model focuses on generating a question based on the target answer within a passage. Several modern NQG models use the Seq2Seq architecture, including RRN, LSTM, and GRU, often incorporating an attention mechanism to process a passage and its target answer. Popular NQG techniques include copying mechanisms and reinforcement learning (Pan, 2019).

- Datasets commonly used in NQG include SQuAD, MS MARCO, NewsQA, RACE, LearningQ, and NarrativeQA.

- The evaluation metrics used in the field include both human and automated techniques such as BLEU, METEOR, ROUGE, and precision.

The remainder of this section will focus primarily on the Seq2Seq architecture (especially GRU), attention mechanisms, lexical features (POS and NER), and evaluation metrics, especially BLEU and precision.

3.1.1 Word Embedding

Language comprehension has always been a strong suit of humans. The relationships between words are often easy for humans to understand; however, this task can be challenging for computers. For example, while humans easily recognize the relationships between words such as "king" and "queen," "man" and "woman," or "tiger" and "tigress," computers must learn to recognize these connections (Yin & Shen, 2018).

Word embeddings are a type of word representation that bridge language understanding between machines and humans. They are n-dimensional text representations in which words with similar meanings are represented by similar vectors that are located close together in vector space. This capability is essential for addressing many challenges in natural language processing.

In word embeddings, each unique word is represented as a real-valued vector in a defined vector space. Each word is characterized by a single vector whose values are learned in a manner similar to a neural network.

Word2Vec (Rong, 2014) is one of the most widely used shallow neural network algorithms for learning word embeddings. It was developed in 2013 by Tomas Mikolov at Google.

3.1.2 Seq2Seq

A major challenge with the basic RNN model is that it struggles with long sentences, often resulting in poor understanding of meaning. To deal with long dependencies, we use sequence-to-sequence (Seq2Seq) models (Shao, 2017).

Deep learning techniques, especially Seq2Seq models, have achieved significant success in applications such as machine translation, text summarization, image captioning, question answering (QA), and question generation (QG). In late 2016, Google Translate began using such a model in its production environment.

A Seq2Seq model generates a new sequence of words from a given input sequence.

The Model As shown in Figure 3.2, the model consists of an encoder and a decoder. Each element in the input sequence is processed by the encoder, which converts the collected data into a vector called the context. Once the entire input sequence has been processed, this context is sent from the encoder to the decoder, which begins to generate the output sequence token by token.

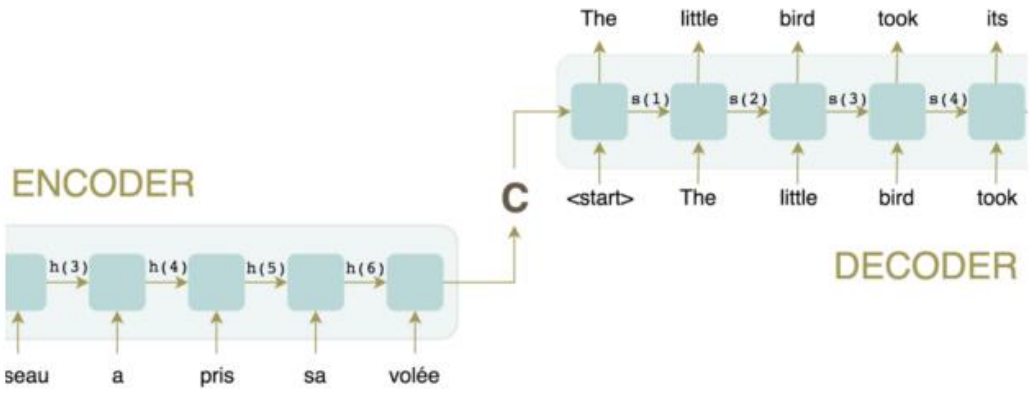


Figure 3.1: [9] Encoder – Decoder architecture.

The context is represented by a vector, and recurrent neural networks (RNNs) are commonly used for both the encoder and decoder. The size of the context vector can be specified during model creation and is typically determined by the number of hidden units in the encoder RNN. In real-world applications, the context vector may be 256, 512, or 1024 units long.

At each time step, an RNN takes two inputs: the current input (in the case of the encoder, a single word from the source sentence) and the previous hidden state. To generate the output for that time step, the RNN combines the current input vector with the previous hidden state. After processing its inputs, the RNN produces an output for that time step and updates its hidden state based on the current and previous inputs. The decoder also maintains hidden states that are carried across time steps, although we haven't illustrated this in this context.

Among the various approaches to sequence-to-sequence modeling, one notable option is the Gated Recurrent Unit (GRU).

3.2 Gated Recurrent Units (GRUs)

GRUs are a special type of RNN designed to learn long-term dependencies. They were introduced in 2014 by Kyunghyun Cho. Like Long Short-Term Memory (LSTM) networks, GRUs manage the flow of information through gates. However, GRUs are relatively new compared to LSTMs, and they often perform better due to their simpler architecture (Yuan, 2019).

The Architecture of the GRU

Now let's understand how GRUs work. A GRU consists of two main gates: the update gate and the reset gate, as shown in Figure 3.3. These gates help determine what information should be retained, passed on, or discarded.

As mentioned earlier, the gates' output values between 0 and 1. A value of 0 indicates that the information is unimportant, while a value of 1 indicates that it is important. Values closer to 0 indicate unimportance and values closer to 1 indicate importance.

At each timestamp t, the GRU takes an input X_t and the previous state H_{t-1} from the previous timestamp. As shown in Figure 3.4, it then generates a new hidden state H_t, which is passed to the next timestamp.

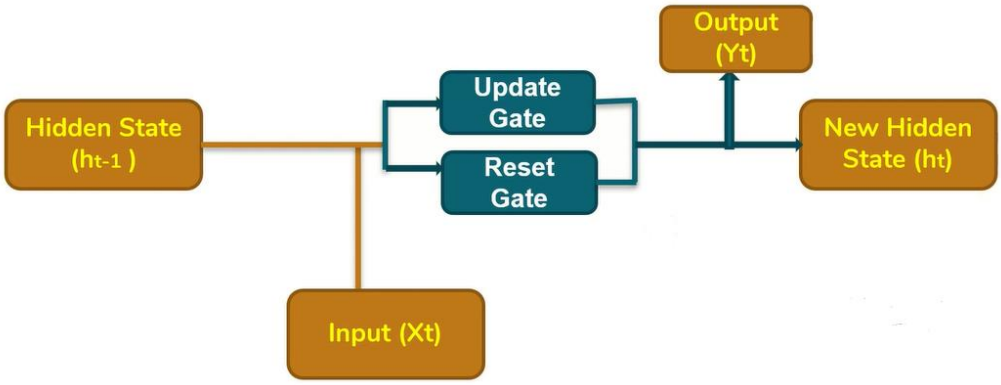


Figure 3.2: Overall structure within the GRU cell (sefidian,2020)

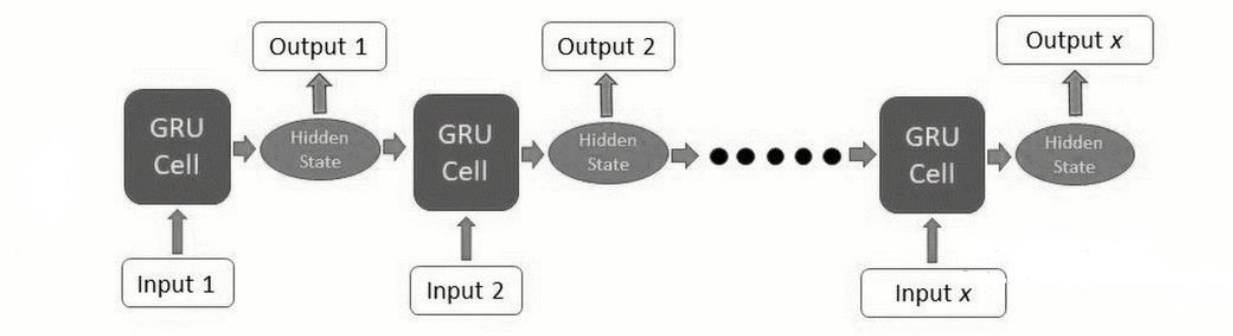


Figure 3.3: GRUs follow the same flow as the typical RNN (sefidian,2020)

Update Gate (z): The primary function of this gate is to inform the model how much of the previous information should be preserved, i.e., passed on to future states.

Reset Gate (r): This gate is used by the model to determine how much information from the past should be forgotten.

As usual, there are weights associated with each gate.

Math and pictorial representation to understand the functioning

Update gate:

- Z_t represents the update gate.
- The parameters are the input representation X_t and the prior hidden state H_{t-1} state information multiplied by their corresponding weights.
- Z_t is calculated using sigmoid activation. as shown in Figure 3.5.

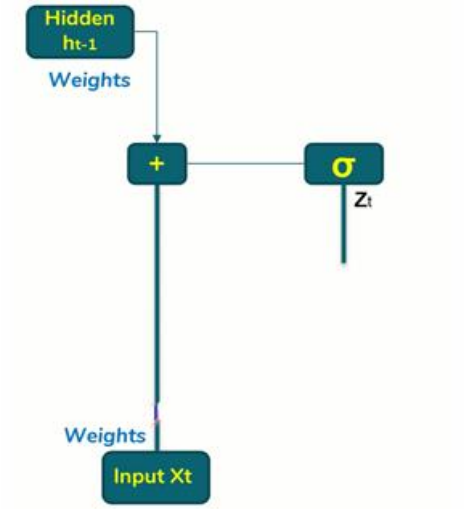


Figure 3.1: Update Gate (andreaperlato, 2022)

$$Z_t = \sigma\left(W^{(Z)} X_t + U^{(Z)} h_{t-1}\right)$$

Where,

- t: current step.
- X_t: Input vector.
- Z_t: update gate vector.
- W and U are vectors and parameter matrices.
- h_{t-1}: The previous hidden state.

Reset gate

- r_t represents the reset gate
- The parameters are the input representation X_t and the prior hidden state H_{t-1} state information multiplied by their corresponding weights.
- r_t is calculated using sigmoid activation as show in Figure 3.6.

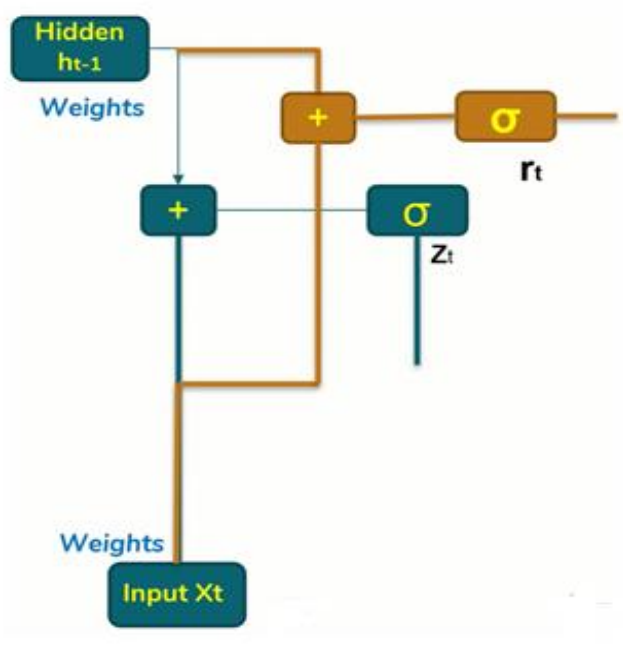


Figure 3.2: Reset Gate (andreaperlato, 2022)

$$r_t = \sigma\left(W^{(Z)} X_t + U^{(r)} h_{t-1}\right)$$

Where,

- t: current step.
- X_t: Input vector.
- r_t: vector of Reset gate.
- W and U: vectors and parameter matrices.
- h_{t-1}: The previous hidden state.

How GRU Works

- A new device has been introduced: the reset gate, which is used to retrieve previously stored data from a memory device.
- Consider a movie review. Initially, you might start with "The movie was directed by X; it starred Y". After about ten lines, you conclude, "I think the movie is bad for the money I paid. In this case, the actual review is the last line. The neural network should not remember the earlier sentences and should focus on the last sentence to capture the essence of your opinion. This focus is enabled by the reset gate.
- To discard irrelevant information, r_t is set to 0 until the last sentence is analyzed.
- Then the tanh activation function is applied, resulting in h'_t (Candidate Hidden State), as shown in Figure 3.7.
- The final phase of the network is to compute and output the h_t vector, which contains information about the current unit.
- This process requires the use of the update gate as shown in Figure 3.8.

$$h'_t = \tanh(W\ X_t + r_t\ \theta\ U\ h_{t-1})$$

$$h_t = Z_t\ \theta\ h_{t-1} + (1 - Z_t)\ \theta\ h'_t$$

Where,

- h'_t: candidate hidden state vector.
- h_t: The output vector.

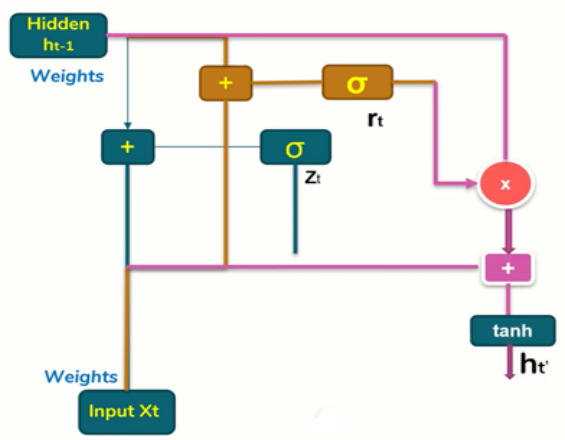


Figure 3.3*:* Candidate hidden state architecture (Krishnan,2022).

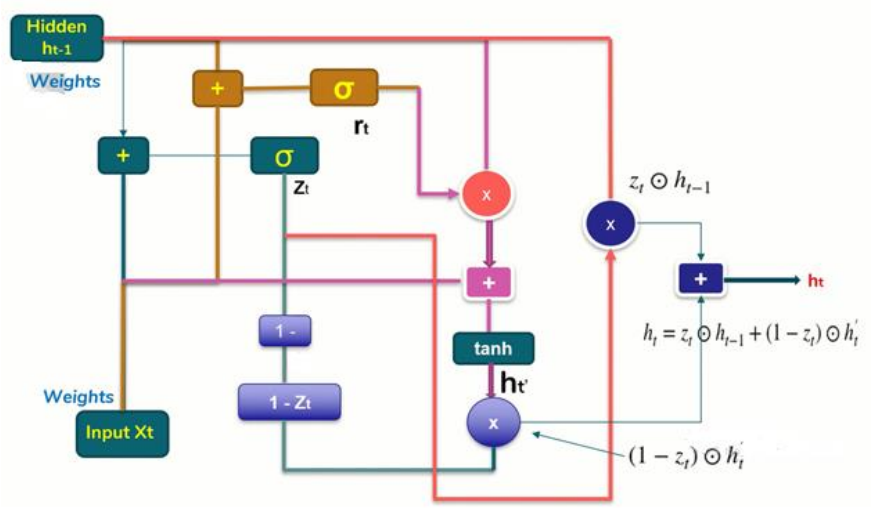


Figure 3.4: The architecture of GRU in recurrent neural networks (Krishnan,2022).

What is the difference between the GRU and the LSTM?

The main differences are as follows (Li, 2021):

- Number of gates: LSTM has three gates, while GRU has two.

- Internal memory and output gate: LSTM contains both an internal memory cell and an output gate, which are absent in GRU.
- Gate functionality: In LSTM, the update gate connects the input and forget gates, while in GRU the reset gate is applied directly to the previous hidden state. In LSTM, the reset gate is shared by the input and forget gates.
- Training parameters: GRU has fewer training parameters than LSTM, which means it uses less memory and runs faster. However, LSTM generally provides more accuracy on large data sets, while it may be less accurate on smaller data sets. If you're working with long sequences and accuracy is critical, LSTM is preferable. If you have limited memory and need faster results, GRU is the better choice.

3.2.1 Word2Vec

Word2Vec is a neural network-based method for rapidly building word embeddings. It was developed by Tomas Mikolov at Google in 2013 in response to the need for more efficient training of neural network-based embeddings, and has since become the de facto standard for developing pre-trained word embeddings.

Word2Vec takes a text document as input and produces a set of feature vectors representing the words in the document. Although Word2Vec is not a deep neural network, it translates text into a numerical representation that deep neural networks can recognize. According to the Word2Vec objective function, words with similar contexts will have similar embeddings. As a result, such words are located close to each other in this vector space. Mathematically, the cosine of the angle Q between these vectors should be close to 1, which means that the angle itself should be as close to 0 as possible, as shown in Figure 3.1. Word2vec has two types: CBOW and the Skip-gram model.

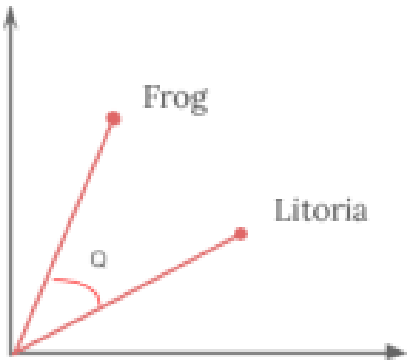


Figure 3.5: Similar words are closely placed in vector space (Great Learning Team, 2020)

3.2.2 Seq2Seq with Attention Mechanism

The Seq2Seq paradigm is designed to transform a source sequence into a target sequence, as shown in Figure 3.9.

When we input an English source sentence into the encoder, it gathers all the information from the source sequence into a single real-valued vector called the context vector. This context vector is then used by the decoder to construct an output sequence in a target language, such as Hindi. The primary goal of the context vector is to condense the entire input sequence into a single representation.

But can a single vector from the encoder effectively contain all the important information when the input sentence is long? Is it possible to predict the target word by focusing on a few relevant words in the sentence rather than relying on a single vector?

The attention mechanism addresses these challenges. Its main purpose is to eliminate the need for a single vector representation for each sentence. Instead, it uses attention weights to focus on specific input vectors from the sentence.

During each decoding step, the decoder receives a set of attention weights that indicate how much "attention" should be given to each input word. These attention weights provide the decoder with contextual information for translation, as shown in Figure 3.10.

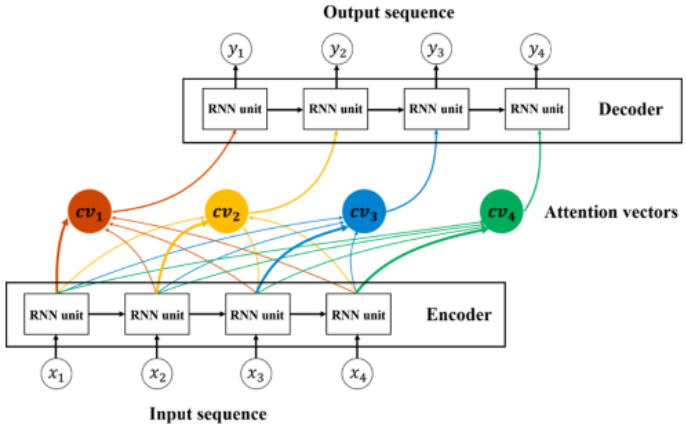


Figure 3.6: Seq2Seq Architecture with attention mechanism (Li,2021).

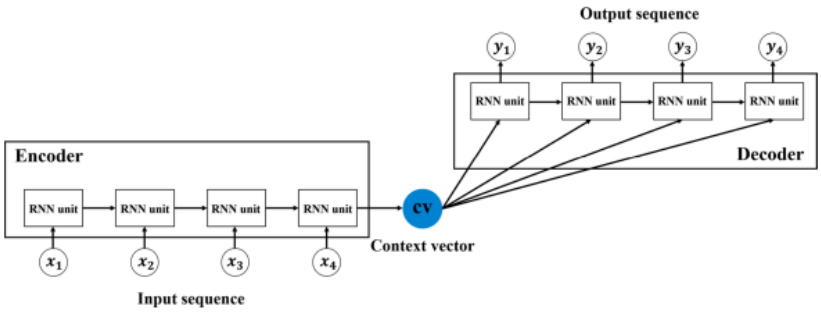


Figure 3.7: Seq2Seq Architecture (Li,2021).

3.2.3 Preprocessing

There are two types of preprocessing: traditional and QG-specific. Traditional preprocessing prepares the input for subsequent tasks, including segmentation, tokenization, and part-of-speech (POS) tagging.

In some cases, named entity recognition (NER) is also required. In this report, we focus on POS tagging and NER.

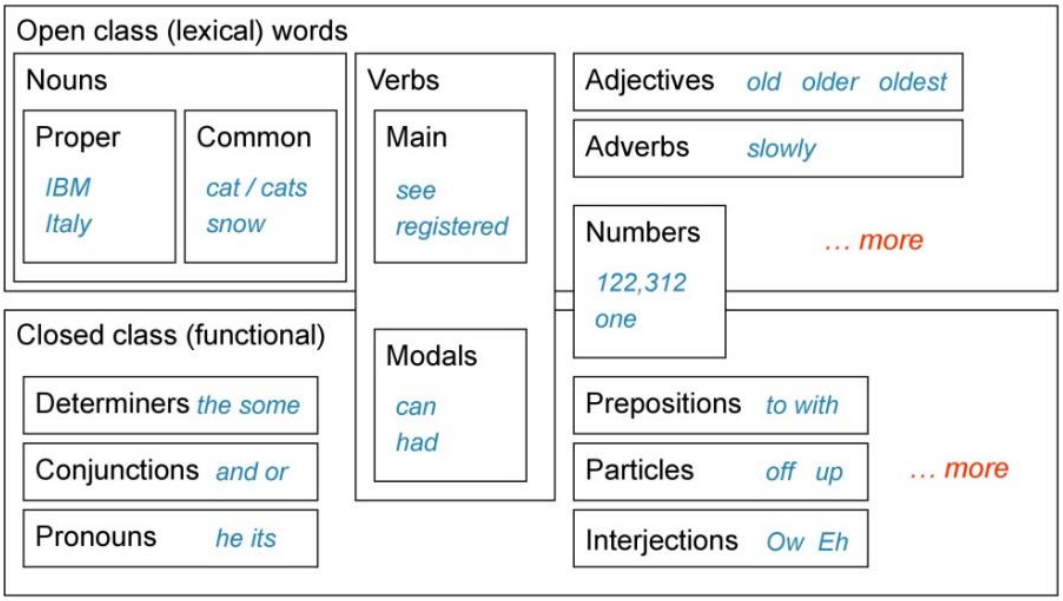
3.2.4 Lexical Feature (POS)

Part-of-speech tagging: Assign a part-of-speech tag to each token in a sentence.
For example:

I	like	his	watch
Pro	verb	pro	noun

the	Fans	watch	the	race
Dt	Noun	verb	dt	noun

The words categorize into 2 classes (open Vs close):



Framework for POS discovering:

- Most freq tag.
- Maxtent P(t/w).
- TnT.
- Bidirectional dependencies.
- Upper bound.

3.2.5 Lexical Features (NER)

Entities: Common things that belong to the noun family.
Named Entity Recognition: A very important subtask to find and classify names in text, for example

Methods of NER

- One approach is to use various machine learning techniques.
- Another option is the Conditional Random Field (CRF), which is supported by both NLP Speech Tagger and NLTK. CRF is a probabilistic model used to model sequential data.
- NER can also be based on deep learning techniques.

3.2.6 BLUE and precision Evaluation Metrics

Precision:

- compared a generated questions to one or more reference questions

$$\text{unigram Precision} = \frac{Num\ word\ matches}{Num\ words\ in\ generation}$$

- But QGS can over generate reasonable words.!!

BLEU, which stands for Bi-Lingual Evaluation Understudy:

- BLEU compared a generated questions to one or more reference questions.
- BLEU compared n-grams of the generation with n-grams of the reference
- BLUE uses a modified n-gram precision to clip the number of matches.

$$\text{Modified unigram Precision} = \frac{clip(Num\ word\ matches)}{Num\ words\ in\ generation}$$

- BLUE also uses bigrams, trigrams, and 4-grams to handle ordering problems

$$4-\text{gram Precision} = \frac{\Sigma clip(Num\ word\ matches)}{\Sigma Num\ 4-grams\ in\ generation}$$

3.3 Update on Research Questions

There has been a significant shift in the research questions. Initially, the focus was on text classification. However, it evolved to Question Generation (QG) because my role as a teaching assistant emphasizes the importance of Natural Question Generation (NQG) for educational purposes.

As a result, we replaced the text classification questions - such as definition, application, limitations, preprocessing, feature extraction, solutions to limitations, datasets used, evaluation metrics, and directions for future research - with NQG questions covering the same areas. The expected responses for each research question are summarized in Table 3.1.

RQ	Expected Outcome / Result
Q1: What is NQG?	It is the task of using deep neural networks to generate questions from a given context.
Q2: What are different applications of NQG?	1- machine reading comprehension 2- Improving question answering system 3- Assisting chatbots in initiating or continuing a conversation with humans
Q3: What are the languages that this research is interested in for NQG?	English Language
Q4: What are limitations and challenges of the existing NQG?	1- Existing neural question generation models are insufficient mostly owing to their failure to adequately simulate the process of how each word in the question is chosen, i.e., whether the text is repeated, or a vocabulary is formed. 2- Most existing solutions are aimed at improving document representations. due to a lack of attention paid to the answer information, The created question may not be appropriate for the answer type, making the response irrelevant.
Q5: What are benchmark datasets to evaluate the performance of models of QAPG?	SQuAD, MS MARCO,NewsQA,RACE,LearningQ and NarrativeQA
Q5.1: What is a proper approach to represent word?	Word Embedding: • Word2Vec • Glove
Q5.2: What are different types of input preprocessing?	There are two forms of preprocessing: traditional preprocessing and QG-specific preparation. Segmentation, phrase splitting, tokenization, POS tagging, and coreference resolution are all part of standard preprocessing, which is used to prepare the data for the next task. In some circumstances, it also entails the recognition of named entities (NER)
Q6: What are the techniques used in NQG?	• NQG models all share the Seq2Seq framework.

	• Or Use seq2seq with attention mechanism
Q7: What are possible Solutions and how to improve the performance of the existing technique of NQG?	• Adding attention mechanism • The model improves when the intended response is used as a guide to help with question generation. Use NLP Tools such as POS and NER.
Q8: What are the methods used to evaluate the performance of models for MQG?	• BLEU, METEOR and ROUGE.
Q9: What are directions for future research on NQG?	• Generation of Deep Questions • Wider Input Modalities • Use reinforcement learning

Table 3.1: The expected result for each research question.

3.4 Threats to Validity

This study has several limitations:

- It focuses solely on the English language.
- It does not address how to predict question types based on input response (e.g., yes/no, multiple choice, or extractive) and context.
- It does not cover the transformer model.
- There is a need for practical applications of the mentioned models to better understand and develop them.

4. CONCLUSION AND FUTURE WORK

This research explores the use of neural network models for generating natural language questions (QG), highlighting their importance for educational materials and for improving question-answering (QA) systems. We analyzed several techniques and evaluation metrics from the literature.

The results show that all NQG models share the Seq2Seq framework. Furthermore, the integration of Seq2Seq with attention mechanisms and the use of part-of-speech (POS) tagging and named entity recognition (NER) contribute to the generation of accurate questions.

The future work of **Question Generation (QG) techniques** focuses on improving the quality, diversity, and applicability of automatically generated questions. Here are some key areas for future research and development:

1. Enhancing Question Quality and Diversity
2. Multimodal Question Generation
3. Personalized and Adaptive QG
4. Integration with Large Language Models (LLMs)
5. Improving QG in Low-Resource Languages
6. Domain-Specific QG
7. Reinforcement Learning (RL) for Question Generation

8. Graph Encoders for Question Generation

5. REFERENCES

Aithal, S. G., Rao, A. B., & Singh, S. (2021). Automatic question-answer pairs generation and question similarity mechanism in question answering system. *Applied Intelligence*, *51*(11), 8484-8497.

Alabbas, W., Al-Khateeb, H. M., & Mansour, A. (2016, October). Arabic text classification methods: Systematic literature review of primary studies. In *2016 4th IEEE International Colloquium on Information Science and Technology (CiSt)* (pp. 361-367). IEEE.

AlKhuzaey, S., Grasso, F., Payne, T. R., & Tamma, V. (2024). Text-based question difficulty prediction: A systematic review of automatic approaches. *International Journal of Artificial Intelligence in Education*, *34*(3), 862-914.

andreaperlato ,2022."Recurrent Neural Network in Theory": aipost , https://www.andreaperlato.com/aipost/recurrent-neural-network-in-theory/

Bennabi, S. R., & Elberrichi, Z. (2020, June). Feature Selection based Arabic Text Classification using Different Machine Learning Algorithms: Comparative Study. In *Proceedings of the 10th International Conference on Information Systems and Technologies* (pp. 1-5).

Bennabi, S. R., & Elberrichi, Z. (2020, June). Feature Selection based Arabic Text Classification using Different Machine Learning Algorithms: Comparative Study. In *Proceedings of the 10th International Conference on Information Systems and Technologies* (pp. 1-5).

Chali, Y., & Hasan, S. A. (2015). Towards topic-to-question generation. *Computational Linguistics*, *41*(1), 1-20.

Du, X., Shao, J., & Cardie, C. (2017). Learning to ask: Neural question generation for reading comprehension. *arXiv preprint arXiv:1705.00106*.

Elbes, M., Aldajah, A., & Sadaqa, O. (2019, October). P-stemmer or NLTK stemmer for arabic text classification?. In *2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS)* (pp. 516-520). IEEE.

Ezzeldin, A. M., & Shaheen, M. (2012, December). A survey of Arabic question answering: challenges, tasks, approaches, tools, and future trends. In *Proceedings of The 13th international Arab conference on information technology (ACIT 2012)* (pp. 1-8).

Heilman, M. (2011). *Automatic factual question generation from text* (Doctoral dissertation, Carnegie Mellon University).

Joseph, S. R., Hlomani, H., Letsholo, K., Kaniwa, F., & Sedimo, K. (2016). Natural language processing: A review. *International Journal of Research in Engineering and Applied Sciences*, *6*(3), 207-210.

Kadhim, A. I. (2019). Survey on supervised machine learning techniques for automatic text classification. *Artificial Intelligence Review*, *52*(1), 273-292.

Krishnan, S., Magalingam, P., & Ibrahim, R. B. (2020). Advanced recurrent neural network with tensorflow for heart disease prediction. International Journal of Advanced Science and Technology, 29(5), 966-977.

Li, A., Xiao, F., Zhang, C., & Fan, C. (2021). Attention-based interpretable neural network for building cooling load prediction. Applied Energy, 299, 117238.

Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M., & Gao, J. (2021). Deep learning--based text classification: a comprehensive review. *ACM Computing Surveys (CSUR)*, *54*(3), 1-40.

Pan, L., Lei, W., Chua, T. S., & Kan, M. Y. (2019). Recent advances in neural question generation. *arXiv preprint arXiv:1905.08949*.

Pandolfi-González, M., Quesada-López, C., Martínez, A., & Jenkins, M. (2020, September). Automatic Classification of Web News: A Systematic Mapping Study. In *Proceedings of SAI Intelligent Systems Conference* (pp. 558-574). Springer, Cham.
Rachid, B. A., Azza, H., & Ghezala, H. H. B. (2020, July). Classification of cyberbullying text in arabic. In *2020 International Joint Conference on Neural Networks (IJCNN)* (pp. 1-7). IEEE.
Reda Affane,2020." Understanding the Hype Around Transformer NLP Models": dataiku, Understanding the Hype Around Transformer NLP Models (dataiku.com)
Rong, X. (2014). word2vec parameter learning explained. *arXiv preprint arXiv:1411.2738*.

Sefidian,2020. "Understanding Gated Recurrent Unit (GRU) with PyTorch code": gated-recurrent-unit-gru-with-pytorch, http://www.sefidian.com/2020/01/30/gated-recurrent-unit-gru-with-pytorch/
Shehab, Mohammed A., et al. "A supervised approach for multi-label classification of Arabic news articles." *2016 7th international conference on computer science and information technology (CSIT)*. IEEE, 2016.
Slobodianiuk, A. V., & Semerikov, S. O. (2025). Advances in neural text generation: A systematic review.
Suhartono, D., Majiid, M. R. N., & Fredyan, R. (2024). Towards automatic question generation using pre-trained model in academic field for Bahasa Indonesia. *Education and Information Technologies*, *29*(16), 21295-21330.
Wahdan, Ahlam, Said A. Salloum, and Khaled Shaalan. "Text Classification of Arabic Text: Deep Learning in ANLP." *International Conference on Advanced Machine Learning Technologies and Applications*. Springer, Cham, 2021
Wahdan, K. A., Hantoobi, S., Salloum, S. A., & Shaalan, K. (2020). A systematic review of text classification research based ondeep learning models in Arabic language. *Int. J. Electr. Comput. Eng*, *10*(6), 6629-6643.
Wang, L., Xu, Z., Lin, Z., Zheng, H., & Shen, Y. (2020, December). Answer-driven Deep Question Generation based on Reinforcement Learning. In *Proceedings of the 28th International Conference on Computational Linguistics* (pp. 5159-5170).
Yang, J., Bai, L., & Guo, Y. (2020, October). A survey of text classification models. In *Proceedings of the 2020 2nd International Conference on Robotics, Intelligent Control and Artificial Intelligence* (pp. 327-334).
Yao, B., Wang, D., Wu, T., Hoang, T., Sun, B., Li, T. J. J., ... & Xu, Y. (2021). It is AI's Turn to Ask Human a Question: Question and Answer Pair Generation for Children Storybooks in FairytaleQA Dataset. *arXiv preprint arXiv:2109.03423*.
Yuan, J., & Tian, Y. (2019). An intelligent fault diagnosis method using GRU neural network towards sequential data in dynamic processes. *Processes*, *7*(3), 152.

Zhang, M. (2021). Applications of deep learning in news text classification. *Scientific Programming*, *2021*.
Zhou, Q., Yang, N., Wei, F., Tan, C., Bao, H., & Zhou, M. (2017, November). Neural question generation from text: A preliminary study. In *National CCF Conference on Natural Language Processing and Chinese Computing* (pp. 662-671). Springer, Cham.
Mucciaccia, S. S., Paixão, T. M., Mutz, F. W., Badue, C. S., de Souza, A. F., & Oliveira-Santos, T. (2025, January). Automatic Multiple-Choice Question Generation and Evaluation Systems Based on LLM: A Study Case With University Resolutions. In *Proceedings of the 31st International Conference on Computational Linguistics* (pp. 2246-2260).
Maity, Subhankar, Aniket Deroy, and Sudeshna Sarkar. "Leveraging In-Context Learning and Retrieval-Augmented Generation for Automatic Question Generation in Educational Domains." *arXiv preprint arXiv:2501.17397* (2025).
Great Learning Team ,2020. "What is Word Embedding | Word2Vec | GloVe".Great learning , https://www.mygreatlearning.com/blog/word-embedding/

Abdeen, M. A., AlBouq, S., Elmahalawy, A., & Shehata, S. (2019). A closer look at arabic text classification. *Int. J. Adv. Comput. Sci. Appl*, *10*(11), 677-688.

Al Faraby, S., Adiwijaya, A., & Romadhony, A. (2024). Review on neural question generation for education purposes. *International Journal of Artificial Intelligence in Education*, *34*(3), 1008-1045.

Al Qadi, L., El Rifai, H., Obaid, S., & Elnagar, A. (2019, October). Arabic text classification of news articles using classical supervised classifiers. In *2019 2nd International conference on new trends in computing sciences (ICTCS)* (pp. 1-6). IEEE.

Kitchenham, B. A. (2012, September). Systematic review in software engineering: where we are and where we should be going. In *Proceedings of the 2nd international workshop on Evidential assessment of software technologies* (pp. 1-2).

Liu, W., Xiao, J., & Hong, M. (2020, January). Comparison on feature selection methods for text classification. In *Proceedings of the 2020 4th international conference on management engineering, software engineering and service sciences* (pp. 82-86).

Maity, S., Deroy, A., & Sarkar, S. (2025). Leveraging In-Context Learning and Retrieval-Augmented Generation for Automatic Question Generation in Educational Domains. *arXiv preprint arXiv:2501.17397*.

Serban, I., Sordoni, A., Bengio, Y., Courville, A., & Pineau, J. (2016, March). Building end-to-end dialogue systems using generative hierarchical neural network models. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 30, No. 1).

Slobodianiuk, A. V., & Semerikov, S. O. (2025). Advances in neural text generation: A systematic review.

Yin, Z., & Shen, Y. (2018). On the dimensionality of word embedding. *Advances in neural information processing systems*, *31*.