Sawaal: A Framework for Automatic Question Generation in Urdu

Maria Rahim and **Shakeel Khoja**
School of Mathematics and Computer Science
Institute of Business Administration
Karachi, Pakistan
{mrkhowaja, skhoja}@iba.edu.pk

Abstract

This study proposes a novel framework for automatic question generation (AQG) designed specifically for the Urdu language. The framework encompasses seven stages, including pre-processing, tagging, anaphora resolution, word chunking, automatic constructed dataset development (ACD) using Urdu linguistic rules, fine-tuning MT5 on various model combinations, and a ranking algorithm. It includes novel algorithms for anaphora resolution and word chunking customized for Urdu's intricate sentence structures. Utilizing these linguistic rules, the framework generates 4,497 question-answer pairs from 250 passages. Additionally, the framework explores fine-tuning Multilingual T5 (MT5) on UQuAD 1.0 (Kazi and Khoja, 2021) and ACD with varied encodings and embeddings, both with and without the proposed algorithms. Generated questions undergo a ranking process based on semantic text representation to ensure relevance and quality, filtering out irrelevant questions. Evaluation using standard automatic metrics such as BLEU-4, METEOR, and ROUGE-L demonstrates the framework's efficacy, with the best-performing model achieving commendable scores of 24.78, 37.07, and 54.99, respectively.

1 Introduction

Automatic question generation (AQG) is an innovative technique that uses artificial intelligence and natural language processing to create questions from textual information. By employing algorithms and language models, AQG can analyze text, understand context, identify key information, and formulate relevant questions. This technology is useful in various domains such as education (Laban et al., 2022), e-learning, content creation, and chatbot systems.

Despite significant advancements in AQG technology for languages like English, there remains a notable research gap in developing an AQG framework specifically for the Urdu language. Multilingual models often fail to capture the unique linguistic, syntactic, and cultural characteristics of individual languages, especially those underrepresented in training datasets, such as Urdu. Urdu's complexity, with characters forming different shapes based on their position in a word and context-sensitive grammar, adds to the challenge (Daud et al., 2017). For instance, for "how many" questions, there can be a variant of کتنے,کتنا,کتنی,کن depending upon the context and grammar of the question. Moreover, the placement of question keywords can change the meaning of the question. For instance, the keyword "کیا" can be used to form yes/no or what questions depending upon its placement in the question. If "کیا" is placed at the start of the sentence and follows the structure of کیا + subject + verb + helping verb, it will generate yes/no questions, for example, (Does it taste good?) “کیااس کاذائقہ اچھاہے؟" . On other hand, if "کیا" is placed in the middle of the sentence it will form a what question, for example, (What is your name?) "آپ کانام کیاہے؟".

The absence of comprehensive frameworks and resources for Urdu limits the availability of automated question generation tools tailored to the specific needs of Urdu-speaking learners, educators, and researchers. To address this research gap, the study proposes a hybrid framework for automatic question generation that combines the linguistic of Urdu with multilingual transformer to generate questions automatically. The major research contribution of this study are:

1. Developed an Automatically Constructed Dataset (ACD) using rule-based approach for Question Generation.

2. Proposed novel framework for anaphora resolution and word chunking.

3. Fine-tuned the multilingual transformer for Urdu language with different embedding, with and without proposed algorithms of anaphora resolution and word chunking.

4. Proposed an algorithm to rank generated question using semantic text representation.

5. Evaluated and compared the result for question generation obtained from each model using automatic and human evaluation.

2 Related Work

Heilman's (Heilman and Smith, 2009, 2010) research established a foundational rule-based framework for Automatic Question Generation (AQG) in English, which has influenced subsequent studies using dependency parsing, POS, and NER tagging, and semantic role labeling with rule-based systems (Khullar et al., 2018; Azevedo et al., 2020; Flor and Riordan, 2018; Dhole and Manning, 2020). Over the past decade, AQG has incorporated machine learning and deep learning techniques, such as RNNs for sequence transduction (Du et al., 2017), reinforcement learning with graph-to-sequence models (Chen et al., 2019), and transformers for faster training (Kriangchaivech and Wangperawong, 2019; Goyal et al., 2024). Notable models include the use of T5 for inferential questions (Ghanem et al., 2022), pretrained BART on an inquisitive dataset (Gao et al., 2022), and a neural question generator trained on diverse datasets (Murakhovs'ka et al., 2022). Additionally, EQG-RACE integrates pre-trained BERT and ELMo embeddings with an Answer-guided Graph Convolution Network (Jia et al., 2021), while another framework uses pre-trained embeddings on BERT and EMLo, trained with GPT and GPT2 (Yuan et al., 2021). A model combining BiLSTM with soft attention and layers for encoding and decoding has also been proposed (Bi et al., 2021). Large language models like GPT-3 and ChatGPT have further advanced AQG with their extensive pre-trained knowledge and sophisticated language understanding (Lee et al., 2023). These language models also support multilingual question generation, but their evaluation in low-resource languages has yet to be explored.

Furthermore, there is a growing interest in extending these capabilities to low-resource languages like Arabic, Hindi, Bengali, which often lack the extensive labelled datasets and advance language processing tools (Kazi et al., 2023).

For the Hindi language, (Anuranjana et al., 2019) proposed a rule-based AQG system utilizing POS tagging, NER tagging, and dependency parsing, enhanced with linguistic rules and IndoWordNet ontology to generate questions. Surface-level and syntactic filters were applied to improve question quality, but these filters sometimes removed important questions containing pronouns for example:
Passage: Nelson Mandela was the first president of South Africa. He was born on 18 July 1918
Question: When was he born?

This question was removed by the filter but "When Nelson Mandela was born?" is the important question. In our proposed frame work, we proposed algorithm for anaphora resolution that replace pronoun with appropriate noun instead of just removing the question with that pronoun. On other hand, (Kumar et al., 2019) proposed a cross-lingual AQG system (CLQG) for Hindi and Chinese, using a shared encoder-decoder architecture trained in two phases: unsupervised training with denoising, autoencoding, and back-translation, followed by supervised training with sequence-to-sequence modeling achieving maximum score of 20.242, 29.143, and 40.643 for BLEU-4, METEOR and ROUGE-L. (Wang et al., 2021) proposed a multilingual language model for automatic question generation in five languages, including Hindi and Chinese, utilizing deep learning models such as Transformer and Multi-BERT achieving highest scores of 35.19 for BLEU-4, 36.25 for METEOR, and 51.23 for ROUGE for Hindi language.

For Arabic, Arabic Question Generator(AQG) claim to be first automatic question generation system since earlier proposed system were semi-automatic (Bousmaha et al., 2020). It combines rule based approach with the semantic role labelling of PropBank (SRL) to generate questions automatically from Arabic text. (Alhashedi et al., 2024) proposed arabic automatic question generation using transformers and scores achieved were 19.12 for BLEU-4, 23.00 for METEOR, and 51.99 for ROUGE-L.

For Bengali language, (Fahad et al., 2024) trained three different answer agnostic transformer model BanglaT5, mT5- base, BanglaGPT2 with different combination of decoding algorithm to generate questions automatically. The scores achieved by their best performing models were 11.42 for BLEU-4, 21.79 for METEOR, and 35.74 for ROUGE-L. On other hand, (Ruma et al., 2023) trained BanglaT5, Mt5-small, Mt5-base transformer model along with the answer for automatic question generation and best model achieved 36.60 Bleu-4, 48.98 METEOR, and 63.38 ROUGE-L scores.

Notably, little to no significant work has been done for AQG in the Urdu language, highlighting a gap in this area of research. To our best knowledge, there is no publication for Urdu language AQG till the writing of this research paper. Hence, to address this gap, the study proposed a hybrid automatic question generation framework that incorporates a rule-based approach with a deep learning model customized for the Urdu language. However, the the framework could be applicable to any language by customizing the rules specific to the language and training the modules of the framework on a corpus specific to that language.

3 Methodology

The proposed framework integrates a rule-based approach with a transformer model and comprises the following seven stages, as seen in Figure 1:

1. Pre-processing
2. Tagging, which includes POS, NER, and dependency parsing
3. Algorithm for Word Chunking
4. Algorithm for Anaphora Resolution
5. Development of Automatically Constructed Dataset (ACD) using rule based approach
6. Fine-tuning multilingual T5 (mT5) model with combination of different embedding and proposed algorithm on ACD and UQuAD 1.0 (Kazi and Khoja, 2021).
7. Ranking of generated questions

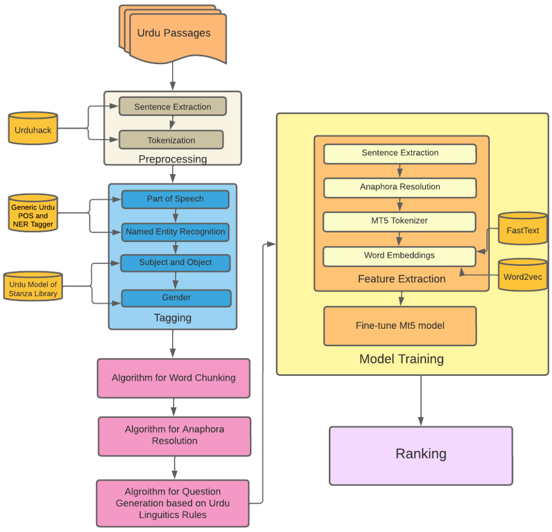

Figure 1: Proposed Framework

3.1 Pre-processing and tagging

Sentence extraction, word tokenization and normalization for each passage are performed using UrduHack (ALi, 2020). The POS are tagged using (Nasim et al., 2020) having F1 score of 96%, NER are tagged using (Kanwal et al., 2019) having F1 score of 77% and the dependency tree is extracted using the Urdu Model of Stanza Library (ALi, 2020).

3.2 Algorithm for Word Chunking

Even after using a word tokenizer, the single word can be treated as multiple words, for example, in the sentence, محمد علی جناح بانی پاکستان ہیں۔ (Muhammad Ali Jinnah is the founder of Pakistan), the word محمد علی جناح (Muhammad Ali Jinnah) is treated as three different words instead one single word. To solve this issue, an algorithm for Urdu noun chunking is developed. The algorithm identifies the group of noun and adjectives that goes together and chucks them together by removing space and putting "-" to form a meaningful word. The steps for the noun chunking algorithm are as follows:

- If a part of speech is followed by the same part of speech, for example, a noun is followed by the same type of noun, they are chunked together. In the sentence above, محمد,علی and جناح are proper nouns (PN), so they are chunked together. However, بانی is a common noun (NN), so it is not chunked together. The chunked word will be محمد-علی-جناح.

- However, chunking the same type of noun is not enough and can result in incorrect chunking, for example, in the sentence محمد علی جناح کراچی میں پیدا ہوئے۔ (Muhammad Ali Jinnah was born in Karachi), محمد-علی-جناح-کراچی (Muhammad Ali Jinnah Karachi) will be chunked together. This is an example of incorrect chunking as کراچی (Karachi) should not be chunked with محمد-علی-جناح (Muhammad Ali Jinnah). To solve this issue, NER tagging is also taken into consideration. The same type of noun is only chunked together if it has the same named entity tag. Since محمد,علی and جناح are tagged as a person, the words will be chunked together, but کراچی will not be chunked with it since its NE tag is location.

- If a noun chunk is preceded by an adjective, the adjective is also chunked with the noun, for example, in the sentence ثناء نے کالی ٹوپی پہن رکھی ہے۔ (Sana is wearing a black cap), the noun ٹوپی(cap) is chuck with the adjective کالی (black).

- If the adjective and noun chunk is preceded by the adverb, the adverb is also chunk with the noun, for example, in the sentence, یہ ایک بہت پرانی عمارت ہے (This is a very old building), the adverb بہت (very), the adjective پرانی (old) and the noun عمارت (building) are chunk together.

- If there is a conjunction between the same type of noun and the named entity tagged, they are chunked together. For example, جناح نے بمبے یونیورسٹی اور لنکن ان لندن سے تعلیم حاصل کی۔ (Jinnah Studied from Bombay university and Lincoln Inn London.), the terms بمبے-یونیورسٹی-اور-لنکن-ان-لندن are chunk together.

In addition to aiding in rule formation, the word chunking algorithm can be utilized to train deep and large language models by using word chunks as answers, particularly in datasets where answers are not available and only passages are provided. The experiments conducted in Section 4 indicate that passing word chunks as answers to mT5 performs better than using answer agnostic transformer.

3.3 Algorithm for Anaphora Resolution

The sentences are extracted from the input passages. Each individual sentence might contain a pronoun referring to a noun in previous sentences. Separating each sentence might result in the pronoun becoming ambiguous. To avoid this ambiguity, a noun and pronoun agreement algorithm has been developed. The algorithm replaces subject pronouns with the corresponding noun from the previous sentence. If the previous sentence does not contain a subject noun, the algorithm continues to look back through earlier sentences until it finds a subject noun, replacing all subject pronouns accordingly. Similarly, object pronouns are replaced with the corresponding object noun from previous sentences, with the last occurring object noun being used to replace the object pronoun in the current sentence.

3.4 Development of ACD

Urdu linguistic rules are employed for the generation of questions from provided passages. In the process, 250 passages were subjected to rule-based Automatic Question Generation (AQG). These passages were sourced from diverse outlets such as Urdu Wikipedia, Urdu stories from Urdu Point, and online Urdu comprehension materials. The resultant dataset, named ACD (Automatically Constructed Dataset), comprises 250 passages with 4497 question-answer pairs. The dataset is , hosted on a private GitHub repository, and it can be requested from by emailing corresponding author at mrkhowaja@iba.edu.pk.The types of questions generated, and the rules applied to generate those questions are as follows:

1. Who Questions (کون/کس)
   کون/کس question words are used depending upon whether the noun is oblique or normative. Oblique nouns are those noun which are followed by preposition whereas normative noun are not followed by preposition. For oblique noun, noun chunk is replaced by کس, for example,چوہدری رحمت علی نے پاکستان کانام رکھا (Chaudhry Rehmat Ali coined the name of Pakistan.) becomes کس نے پاکستان کانام رکھا؟. For normative noun, noun chunk is replaced کون, for example, جارج واشنگٹن امریکہ کے پہلے صدر تھے (George Washington was the first president of the United States) becomes کون امریکہ کے پہلے صدر تھے؟

2. What Questions (کیا)
   If the word chunk is the object noun, it is replaced by کیا, for example, in sentence بچہ رنگین تصویر بنارہا ہے۔ (The child is drawing a colorful picture.) the رنگین تصویر (colorful picture) is replaced by کیا

3. Where Questions (کہاں)
   If the chunked noun is a location or organization, it is replaced by کہاں. For oblique noun, only chunk noun is removed for example, جناح نے بمبے یونیورسٹی اور لنکن ان لندن سے تعلیم حاصل کی (Jinnah studied at Bombay University and Lincoln’s Inn, London.) becomes جناح نے کہاں سے تعلیم حاصل کی؟ (Where did Jinnah studied from?). For normative noun, chunk noun along with its preposition is removed, for example سارہ کراچی میں رہتی ہے (Sarah lives in Karachi) becomes سارہ کہاں رہتی ہے؟ (Where Sarah lives?)

4. When Questions (کب)
   Dates or times found by the NER tagger are replaced by کب along with their prepositions. For example: محمد علی جناح25دسمبر1876کو پیدا ہوئے (Muhammad Ali Jinnah was born on 25 December, 1876.) becomes محمد علی جناح کب پیدا ہوئے؟ (When was Muhammad Ali Jinnah born?)

5. How Many Questions (کیتنی/کیتنے)
   In Urdu, the choice between کیتنی/کیتنے for ”How many” questions and کتنا for ”How much” questions depends on whether the noun is countable or uncountable. For instance, in the question کتناوقت لگے گا؟ (How much time will it take?), وقت (time) being uncountable uses کتنا (How much). However, Urdu lacks a noun tagger to distinguish countable and uncountable nouns, limiting the study to ”How many” questions. When a cardinal value is identified, it is replaced with either کیتنی or کیتنے based on the gender of the dependent noun. For example, in the sentence کلاس میں 10لڑکیاں ہیں۔ (There are 10 girls in the class), لڑکیاں (girls) being feminine, is replaced by کتنی, forming the question کلاس میں کتنی لڑکیاں ہیں؟ (How many girls are there in the class?). Similarly, in کلاس میں10لڑکے ہیں۔ (There are 10 boys in the class), لڑکے being masculine is replaced by کیتنے, forming کلاس میں کتنے لڑکے ہیں؟ (How many boys are there in the class?). When both masculine and feminine nouns are involved, کیتنے is used, as in کلاس میں10لڑکے اور لڑکیاں ہیں۔ (There are 10 boys and girls in the class), resulting in کلاس میں کتنے لڑکے اور لڑکیاں ہیں؟ (How many boys and girls are there in the class?). For oblique nouns like in ثنااس فرم میں10سال سے کام کررہی ہے۔ (Sana has been working in this firm for 10 years), سال (years), followed by the preposition سے, uses کیتنے, resulting in ثنااس فرم میں کیتنے سال سے کام کررہی ہے؟ (How many years has Sana been working in this firm?).

6. Why Questions (کیوں)
   The کیوں is used to ask why Question. When a sentence contains کیونکہ (because), the whole chunk after کیونکہ along with کیونکہ is removed and the word کیوں is placed at the beginning of the sentence. For example, in the sentence محمد علی جناح نے کانگریس چھوڑدی کیونکہ سیاسی اختلاف تھا (Muhammad Ali Jinnah left congress because of political difference.), the whole chunk کیونکہ سیاسی اختلاف تھا is removed and کیوں is placed at the beginning of the sentence. The question formed will be کیوں محمد علی جناح نے کانگریس چھوڑدی؟

The question generated were evaluated by human evaluator on the basics of syntax, semantics and relevance on 10-likert scale. Question with average accuracy of less than 50% on any of these metrics were removed. The final set of data of 250 passages and 4497 questions as seen in Table 1 and distribution of types of questions can be seen in Figure ??.

Metrics	Value
Total Passages	250
Total Questions	4497
Average Length of Question	14
Average Length of Paragraph	9 Sentences
Types of Questions	6
Maximum Length of Question	31
Number of Who Questions	989
Number of What Questions	855
Number of When Questions	900
Number of How many Questions	719
Number of Why Questions	90

Table 1: ACD Statistics

3.5 Automatic Question Generation using Multilingual Transformers

In this framework, a text-to-text transformer architecture is proposed for automatic question generation, favoring transformers over models like RNNs and LSTMs due to their efficiency in capturing long-range dependencies and faster training speed. This efficiency is attributed to the self-attention mechanism in transformers, allowing each token to attend to all others in the sequence. We employed Multilingual T5 (MT5) as it integrates both encoder and decoder models trained on various languages. This architecture allows us to encode context (passage) and answer and decode generated question as the output. The sample output is as follows:

Context:

جناح نے 1913 سے 14 اگست 1947 کو پاکستان کے قیام تک آل انڈیا مسلم لیگ کے
رہنما کے طور پر خدمات انجام دیں، اور پھر اپنی موت تک پاکستان کے پہلے گورنر جنرل کے
ڈومینین کے طور پر خدمات انجام دیں۔

Answer:

جناح

Questions:

کون 1913 سے 1947 تک آل انڈیا مسلم لیگ کے رہنما تھے؟
کون 1913 سے 1947 تک آل انڈیا مسلم لیگ کے سربراہ تھے؟
کس نے 1913 سے 1947 تک آل انڈیا مسلم لیگ کی قیادت کی؟

To enhance context utilization and address computational constraints, the MT5 model is trained at the sentence level rather than the passage level. Sentences are extracted and anaphora resolution is applied to resolve ambiguous references, followed by tokenization using sentence piece. To capture similar words, Word2Vec and FastText word embeddings are employed. The encoder maps each word in a sentence to a dimensional vector, and sentences are padded to a maximum length of 512 for uniformity. The self-attention mechanism in the encoder captures dependencies between words in both local and global contexts. Input and target sequences are formatted and passed to the transformer for output generation. The decoder generates the output sequence using self-attention, with beam search employed to explore multiple candidate sequences simultaneously, enhancing the likelihood of capturing important and diverse questions. For example, consider the Urdu sentence, ہمالیہ میں واقع ماؤنٹ ایورسٹ دنیا کا سب سے اونچا پہاڑ ہے۔(Mount Everest in the Himalayas is the highest mountain in the world.) with answer ہمالیہ (Himalayas). The greedy search made the following question دنیا کا سب سے اونچا پہاڑ کس میں واقع ہے؟ (Which is the highest mountain in the world located?) but the top questions using beam search were ماؤنٹ ایورسٹ کہاں واقع ہے؟ (Where is Mount Everest Located?), دنیا کا سب سے اونچا پہاڑ کہاں واقع ہے؟ (Where is the highest mountain in the world located?) and دنیا کا سب سے اونچا پہاڑ کہاں پایا جا سکتا ہے؟.(Where can the highest mountain in the world be found?) The snapshot of the probability calculated using beam search can be seen in Figure 2.

3.6 Ranking of Generated Questions

Multiple questions can be generated from a single sentence which results in an over generation of questions. To resolve this issue, the questions are ranked, and the top 10 questions are selected. Currently, the ranking algorithm is determined by their similarity to the original sentence. Sentence embeddings for each sentence in the paragraph are calculated using SBert multilingual. Subsequently, the embeddings of each sentence in the passage are averaged to create the passage's overall embedding. Likewise, sentence embeddings for the questions are computed. After which, cosine similarity is employed to measure the similarity between the passage and each question, and the top 10 questions with the highest similarity scores are selected as seen in Figure 3.

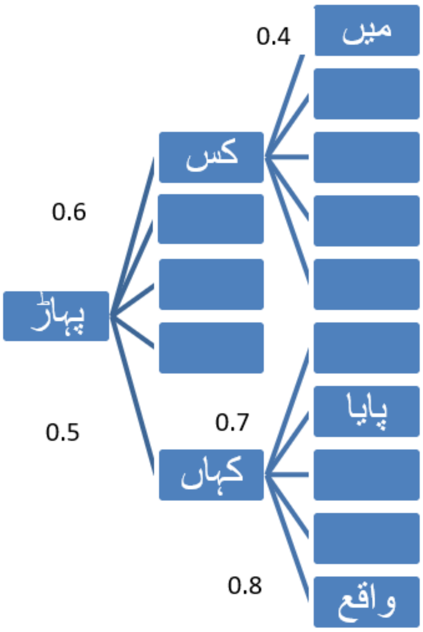

Figure 2: Probability Distribution of Beam Search Candidates

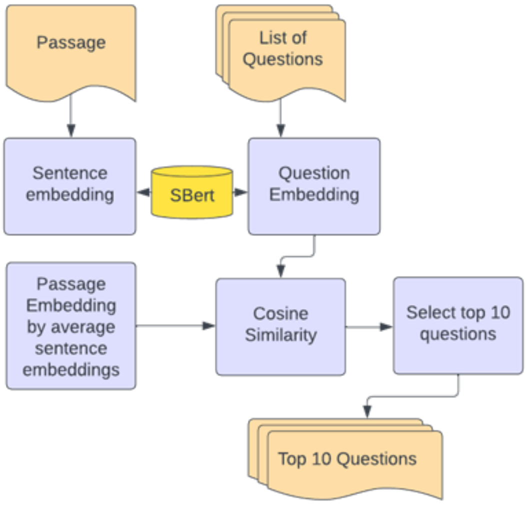

Figure 3: Proposed Ranking Algorithms for Generated Questions

4 Experiment

The study conducted four different sets of experiments on the ACD and UQuAD 1.0 (Kazi and Khoja, 2021) datasets , utilizing pre-trained embeddings, specifically Urdu word embeddings (Haider, 2018) and FastText (Grave et al., 2018), integrated with a deep learning model both with and without anaphora resolution and a ranking algorithm. Additionally, various hyper-parameters such as learning rate, number of epochs, and batch size were systematically adjusted to optimize the model's performance. The model performed best with the hyper parameters shown in Table 2, considering the computational limitations.

Hyperparameter	**Value**
Epochs	3
Optimizer	Adam
Batch Size	12
Learning Rate	1e-5
Number of Beams	5
Number of Sequence	3

Table 2: Optimal Hyper parameters for Model Performance

- Experiment 1: Fine-tuned Mt5 on ACD with Urdu word embeddings (Haider, 2018) and FastText (Grave et al., 2018).
- Experiment 2: Fine-tuned Mt5 on ACD along with two specified embeddings and anaphora resolution named Sawaal.
- Experiment 3: Fine-tuned Sawaal along with ranking algorithm.
- Experiment 4: Fine-tuned best performing model i.e. MT5 along with anaphora resolution and ranking algorithm on FastText embeddings on the following dataset and encoding:
  1. Answer Aware MT5 trained on ACD.
  2. Answer Agnostic MT5 trained on ACD.
  3. Word Chunks encoded as answer for MT5 trained on ACD.
  4. Answer Aware MT5 trained on UQuAD 1.0 (Kazi and Khoja, 2021).
  5. Answer Agnostic MT5 trained on UQuAD 1.0 (Kazi and Khoja, 2021).
  6. Word Chunks encoded as answer for MT5 trained on UQuAD 1.0 (Kazi and Khoja, 2021).

The experiments were designed to analyze the accuracy of the proposed anaphora resolution and ranking algorithms. By comparing these experiments, the study aimed to gain insights into the effectiveness of different word

Model	BLEU-4	METEOR	ROUGE-L
MT5 + Urdu embeddings	17.73	20.97	39.89
Sawaal + Fast Text embeddings	21.36	35.23	52.38
Sawaal + Urdu embeddings	22.14	34.02	51.65
Sawaal + Fast Text embeddings + Ranking	24.78	37.07	54.99
Sawaal + Urdu embeddings + Ranking	23.32	36.47	53.66

Table 3: Evaluation Metrics for Question Generation Models

embedding techniques and the proposed algorithms when combined with large learning models like MT5. Specifically, the comparison between experiment 1 and experiment 2 highlights the impact of the anaphora resolution algorithm on improving model accuracy. The comparison between experiment 2 and experiment 3 evaluates the effectiveness of the ranking algorithm. Finally, experiment 4 assesses the model's adaptability and generalization capabilities when trained on different datasets, while also evaluating the efficiency of using word chunks for answer encoding within the MT5 framework.

5 Evaluation

The questions produced by the framework undergo evaluation against the UQuAD 1.0 test dataset to compute metrics such as F-scores for METEOR (Banerjee and Lavie, 2005), BLEU-4(Papineni et al., 2002), and ROUGE-L (Lin, 2004). Table 3 presents the scores achieved by each model in the experiment 1-3, utilizing the following encoding format:

input = ”context: %s answer: %s </s>” %
target = ”question: %s </s>” %

While Table 4 and Figure 4 presents the scores of fine-tuning the T5 model on different datasets and combination of various input encoding method.

Combinations	Datasets	Encoding
1	UQuAD 1.0	(p,a)
2	UQuAD 1.0	(p)
3	UQuAD 1.0	(p,wc)
4	ACD	(p,a)
5	ACD	(p)
6	ACD	(p,wc)

Table 4: Encoding Combinations for Datasets. p stands for paragraph, a stands for answer and wc stands for chunks

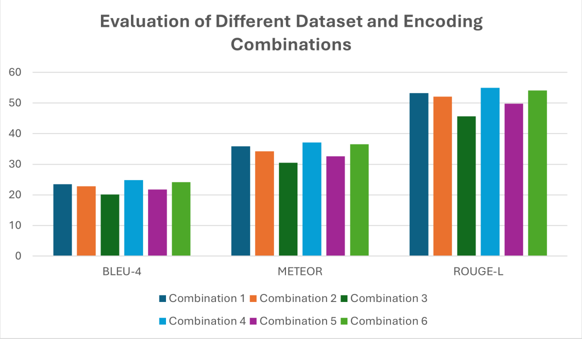

Figure 4: Evaluation of Different Dataset and Encoding Combinations

The study also evaluated final set of generated question from 10 human expert who evaluated the question based on syntax, relevance and semantics of the question on 10-likert scale. The average score for syntax, semantics and relevance were as 8.4, 8.2 and 7.7 respectively.

Limitations

While the proposed framework is able to generate semantically, syntactically and relevant questions from the passage it also have few drawbacks. Firstly, both datasets used for model training contain errors. UQuAD 1.0 contains translation errors, while ACD suffers from errors due to rule exceptions, insufficient language processing tools, incorrect tagging. Secondly, in the studies only pre-trained word embeddings are used. Future research aims to train conceptual embedding such as mBERT and SBERT for urdu language to enhance semantic understanding and context in natural language processing tasks. Thirdly, current system limitations include treating all sentences equally in importance and relying solely on similarity measures for question ranking, without considering whether the answer is present in the passage, which are aspects intended for future implementation.

Ethics Statement

This study adheres to the highest ethical standards in research. All data used, including text passages and question-answer pairs, were sourced from publicly available materials and databases, ensuring that no private or sensitive information was used without explicit consent. Additionally, all sources of data have been properly cited, and the use of copyrighted materials complies with relevant laws and guidelines.

References

Saleh Saleh Alhashedi, Norhaida Mohd Suaib, and Aryati Bakri. 2024. Arabic automatic question generation using transformer model. In *AIP Conference Proceedings*, volume 2991. AIP Publishing.

Ikram ALi. 2020. Urduhack: A python library for urdu language processing.

Kaveri Anuranjana, Vijjini Anvesh Rao, and Radhika Mamidi. 2019. Hindi question generation using dependency structures.

Pedro Azevedo, Bernardo Leite, Henrique Lopes Cardoso, Daniel Castro Silva, and Luís Paulo Reis. 2020. Exploring nlp and information extraction to jointly address question generation and answering. In *IFIP International Conference on Artificial Intelligence Applications and Innovations*, pages 396–407. Springer.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

Sheng Bi, Xiya Cheng, Yuan-Fang Li, Lizhen Qu, Shirong Shen, Guilin Qi, Lu Pan, and Yinlin Jiang. 2021. Simple or complex? complexity-controllable question generation with soft templates and deep mixture of experts model. *arXiv preprint arXiv:2110.06560*.

Kheira Zineb Bousmaha, Nour H Chergui, Mahfoud Sid Ali Mbarek, and Lamia Hadrich Belguith. 2020. Aqg: Arabic question generator. *Rev. d'Intelligence Artif.*, 34(6):721–729.

Yu Chen, Lingfei Wu, and Mohammed J Zaki. 2019. Reinforcement learning based graph-to-sequence model for natural question generation. *arXiv preprint arXiv:1908.04942*.

Ali Daud, Wahab Khan, and Dunren Che. 2017. Urdu language processing: a survey. *Artificial Intelligence Review*, 47:279–311.

Kaustubh D Dhole and Christopher D Manning. 2020. Syn-qg: Syntactic and shallow semantic rules for question generation.

Xinya Du, Junru Shao, and Claire Cardie. 2017. Learning to ask: Neural question generation for reading comprehension.

Abdur Rahman Fahad, Nazme Al Nahian, Md Ahanaf Islam, and Rashedur M Rahman. 2024. Answer agnostic question generation in bangla language. *International Journal of Networked and Distributed Computing*, pages 1–26.

Michael Flor and Brian Riordan. 2018. A semantic role-based approach to open-domain automatic question generation. In *Proceedings of the thirteenth workshop on innovative use of NLP for building educational applications*, pages 254–263.

Lingyu Gao, Debanjan Ghosh, and Kevin Gimpel. 2022. "what makes a question inquisitive?" a study on type-controlled inquisitive question generation. In *Proceedings of the 11th Joint Conference on Lexical and Computational Semantics*, pages 240–257, Seattle, Washington. Association for Computational Linguistics.

Bilal Ghanem, Lauren Lutz Coleman, Julia Rivard Dexter, Spencer von der Ohe, and Alona Fyshe. 2022. Question generation for reading comprehension assessment by modeling how and what to ask. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2131–2146, Dublin, Ireland. Association for Computational Linguistics.

Rupali Goyal, Parteek Kumar, and VP Singh. 2024. Automated question and answer generation from texts using text-to-text transformers. *Arabian Journal for Science and Engineering*, 49(3):3027–3041.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning word vectors for 157 languages. *arXiv preprint arXiv:1802.06893*.

Samar Haider. 2018. Urdu word embeddings. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Michael Heilman and Noah A Smith. 2009. Question generation via overgenerating transformations and ranking. Technical report, Carnegie-Mellon Univ Pittsburgh pa language technologies insT.

Michael Heilman and Noah A Smith. 2010. 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*, pages 609–617.

Xin Jia, Wenjie Zhou, Xu Sun, and Yunfang Wu. 2021. Eqg-race: Examination-type question generation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 13143–13151.

Safia Kanwal, Kamran Malik, Khurram Shahzad, Faisal Aslam, and Zubair Nawaz. 2019. Urdu named entity recognition: Corpus generation and deep learning applications. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 19(1):1–13.

Samreen Kazi and Shakeel Khoja. 2021. Uquad1. 0: Development of an urdu question answering training data for machine reading comprehension.

Samreen Kazi, Shakeel Khoja, and Ali Daud. 2023. A survey of deep learning techniques for machine reading comprehension. *Artificial Intelligence Review*, 56(Suppl 2):2509–2569.

Payal Khullar, Konigari Rachna, Mukul Hase, and Manish Shrivastava. 2018. Automatic question generation using relative pronouns and adverbs. In *Proceedings of ACL 2018, Student Research Workshop*, pages 153–158.

Kettip Kriangchaivech and Artit Wangperawong. 2019. Question generation by transformers.

Vishwajeet Kumar, Nitish Joshi, Arijit Mukherjee, Ganesh Ramakrishnan, and Preethi Jyothi. 2019. Cross-lingual training for automatic question generation.

Philippe Laban, Chien-Sheng Wu, Lidiya Murakhovs'ka, Wenhao Liu, and Caiming Xiong. 2022. Quiz design task: Helping teachers create quizzes with automated question generation. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 102–111, Seattle, United States. Association for Computational Linguistics.

Unggi Lee, Haewon Jung, Younghoon Jeon, Younghoon Sohn, Wonhee Hwang, Jewoong Moon, and Hyeoncheol Kim. 2023. Few-shot is enough: exploring chatgpt prompt engineering method for automatic question generation in english education. *Education and Information Technologies*, pages 1–33.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Lidiya Murakhovs'ka, Chien-Sheng Wu, Philippe Laban, Tong Niu, Wenhao Liu, and Caiming Xiong. 2022. MixQG: Neural question generation with mixed answer types. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1486–1497, Seattle, United States. Association for Computational Linguistics.

Zarmeen Nasim, Shaukat Abidi, and Sajjad Haider. 2020. Modeling pos tagging for the urdu language. In *2020 International Conference on Emerging Trends in Smart Technologies (ICETST)*, pages 1–6. IEEE.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Jannatul Ferdous Ruma, Tasmiah Tahsin Mayeesha, and Rashedur M Rahman. 2023. Transformer based answer-aware bengali question generation. *International Journal of Cognitive Computing in Engineering*, 4:314–326.

Bingning Wang, Ting Yao, Weipeng Chen, Jingfang Xu, and Xiaochuan Wang. 2021. Multilingual question generation with language agnostic language model. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2262–2272.

Wei Yuan, Tieke He, and Xinyu Dai. 2021. Improving neural question generation using deep linguistic representation. In *Proceedings of the Web Conference 2021*, pages 3489–3500.